The Transcriptomic Landscape of HIV-TB

Thesis Presented for the Degree of

DOCTOR OF PHILOSOPHY

in the Department of Medicine

UNIVERSITY OF CAPE TOWN

Armin Deffur

25th November 2013

This work is dedicated to Viliana, who has supported me throughout this venture through some very challenging circumstances, and has taught me the value of balance between work- and home life, and to Alexander, who constantly reminds me of the value of play in discovering the world.

Acknowledgements

First and foremost I wish to thank my sponsor and co-supervisor for this PhD degree, Professor Robert J Wilkinson, for his unwavering support. In his words, I am "an infectious disease physician who writes equations for fun" (there are six in this thesis!). This pretty much sums up a core difficulty for me: where do I, with my interests that range from clinical medicine to computer science actually fit in? The answer is clear: in a research group that is open to, and indeed requires, interdisciplinary approaches in problem solving. My current working environment is the embodiment of this: where else could a clinician develop several thousand lines of R code in order to address a complex biomedical problem like HIV-TB? I trust that this work will justify the investment that Professor Wilkinson was willing to make.

No study can succeed without access to willing study participants. I am extremely fortunate to have had access to a treasure trove of prior data as well as ongoing study subject recruitment and data collection in the context to the long-running IMPI-Africa registry, headed by my co-supervisor and Head of Department, Professor Bongani M Mayosi. Under often difficult conditions, a large dataset that is unique in the world has been steadily accruing. This thesis examines individuals studied in the context of IMPI-Africa, but does not concentrate on the cardiac aspects of the condition, but rather takes advantage of sampling the site of tuberculosis disease.

Development of an analytic strategy for heterogeneous data is tricky at best, and a nightmare at worst. In order to figure out how to approach the IMPI-MA data (a total of approximately 8.9 *million* probe intensity values) I often relied on sage advice of my main thesis supervisor, Professor Nicola Mulder. My initially lofty, if nebulous analysis goals were slowly but surely converted to practical ideas that I could implement in code. Knowing what to leave out is often as important as deciding what to include, and Prof. Mulder was of great assistance in this regard.

Absolutely vital to the success of this project was the collaboration with Dr Anne O'Garra's laboratory a the National Institute for Medical Research, London, UK. Building on her existing collaboration with Professor Robert J Wilkinson, I was able to access the expertise in the O'Garra laboratory in the following crucial areas: study design, the choice of negative controls, assessment of RNA quality and preparation of RNA samples for microarray analysis, as well as approaches to data interpretation. Many thanks go to Dr Christine Graham in the O'Garra laboratory for single-

handedly processing my extracted RNA samples and preparing the sample library for microarray. Thanks to her expert contribution we ended up with a very high quality dataset. Discussions with Dr O'Garra and Dr Chloe Bloom were very helpful in shaping analytic approaches. Finally, Harsha Jani in the shared microarray facility at the NIMR hybridised and imaged all arrays. Additional thanks to her for re-scanning my samples so that I could get my hands on the original TIFF image data and bead-level data!

Virtually all of science today relies on work that has gone before, and in this context I would like to thank Kerryn Matthews, who took on the immunology of tuberculous pericarditis several years ago together with her supervisor, Professor Katalin A Wilkinson, and generated a huge amount of data whose value is yet to be fully realised. Many RNA samples that went on to microarray were collected and processed by Kerryn. Mpiko Ntsekhe, now Professor and Head of the Division of Cardiology in the Department of Medicine at UCT, together with his colleagues, contributed a very large proportion of the phenotypic data that enabled me to assign samples to classes and which provides a very rich physiological context for the various samples. Without this phenotypic data many questions could not have been asked.

For the prospective sample collection I relied heavily on the IMPI team, both in the CathLab and the IMPI Research Office. Thank you to Veronica Francis, Unita September, and Sisters Hartnick, Williams, Hare and Bing and all the Cardiology Fellows for your efforts in ensuring that I was able to obtain these precious samples!

A fair amount of laboratory time was required to process the prospectively collected samples. Many thanks to Ronnett Seldon for her help in the BSL-3 facility, and Nzwaki Bangani for her help in extracting the RNA.

Many thanks to Rene Goliath and Kathryn Wood for expert logistical and emotional support. Without their input this project would have been a non-starter.

The R project and its sibling, the bioconductor project, as well as the various projects contributing to the TEX/ LATEX codebase provide us with free software. The fact this this software is free is no way a reflection of its value, rather the opposite is true. R is at the cutting edge of data analytic software, and its role will continue to grow in this era of "Big Data". In this context I would like to thank Renaud Gaujoux and Cathal Seoighe for contributing the CellMix package, which was key in one of three analytic steps applied to the microarray data, and of course all the unnamed developers

who have contributed to the open source code base.

This project could not have been accomplished without adequate funding. I wish to thank all funders for their vital contributions to this project:

1. The Clinical Infectious Disease Research Initiative, Institute for Infectious Diseases and Molecular Medicine (via Wellcome Trust)
2. Southern African Consortium for Research Excellence (via Wellcome Trust)
3. Medical Research Council of South Africa
4. Hasso Plattner Foundation
5. Harry Crossley Foundation
6. Faculty of Health Sciences, University of Cape Town

Last but not least I again thank Viliana for being there for me all the way, supporting my dream as well as myself, and my son Alexander, who, while younger than this research project, continues to learn and innovate on a daily basis at such a pace that I feel humbled.

Abstract

The Transcriptomic Landscape of HIV-TB

Armin Deffur

August 2013

The thesis consists of four main parts. In Part 1 of this thesis I provide a broad overview of HIV-TB with an emphasis on systems approaches followed by an overview of a systems-level study aiming at addressing hypotheses relating to transcriptional differences in active tuberculosis and HIV-1 infection, measured in blood and the site of disease in tuberculous pericarditis. The final chapter in this part describes the methods used to generate and analyse the systems-level data, with emphasis on microarray data generation and analysis.

Part 2 first presents analysis of transcriptional data generated by RT-PCR at the site of disease compared to blood in study subjects with tuberculous pericarditis, with results showing clear evidence of transcriptional differences between compartments. A Technical Results chapter then provides an overview of the microarray data, and an analytic paradigm based on sample embedding in high-dimensional phenotype space is developed. I then assess the overall quality of the dataset and exclude large-scale systematic bias, while comparison of the IMPI-MA data to exiting TB transcriptomic data shows a close match. Part 2 concludes with a description of a comprehensive analytic framework developed for the IMPI-MA data.

Part 3 presents the results of the analytic pipeline as applied to the transcriptional response in blood to active tuberculosis in an HIV-1 uninfected population. A signature of active tuberculosis is described, and deconvolution analysis finds significant NK cell activation in active tuberculosis. Cell-type specific differential expression identifies CD4 T cells, NK cells and neutrophils as the most likely contributors to the overall "signature" of active tuberculosis. Weighted gene co-expression network analysis reveals multiple modules, whose expression is shown to be differentially regulated based on disease category.

Part 4 summarises the results of analysing contrasts in three main contexts: tuberculosis, HIV-1 infection and compartment. Two novel results are presented: Firstly, NK cells are shown to

be functionally downregulated at the site of disease, suggesting a possible defence mechanism by *M. tuberculosis*, and secondly, large-scale metabolic pathway dysregulation at the site of disease, possibly favouring *M. tuberculosis*, is demonstrated.

Part 5 concludes the thesis with a summary and outlines future work.

Contents

Terminology and abbreviations 25

I Introduction and methods **27**

1 HIV/TB and studies at the site of disease **29**

1.1 Introduction/outline 29

1.2 Clinical features and importance of HIV-TB 30

1.3 Clinical phenotypes and diagnosis: HIV-TB is a heterogeneous syndrome 32

1.4 Immune reconstitution after ART 34

1.5 Disease mechanisms in co-infection 35

1.6 Experimental systems 37

1.7 Systems approaches 41

1.8 Conclusion 49

2 IMPI-MA: a two-compartment survey of the transcriptional landscape in HIV-TB **51**

2.1 Tuberculous pericarditis 51

2.2 IMPI and related substudies 53

2.3 Development of a systems approach 56

2.4 Hypotheses and study questions 57

2.5 Aims and objectives 57

2.6 Overall study design 58

3 Materials and methods **61**

3.1 Preliminary data: RT-PCR data generation and analysis 61

3.2 IMPI-MA study subjects . . . 65
3.3 RNA and other Samples . . . 70
3.4 Clinical phenotype database . . . 73
3.5 RNA processing for microarray . . . 77
3.6 Microarray workflow . . . 78
3.7 Microarray data . . . 79
3.8 Data analysis overview . . . 82
3.9 Code descriptions . . . 96
3.10 Comparisons . . . 100

II General Results **101**

4 RT-PCR on matched blood and fluid **103**
4.1 General comments regarding the presentation of results . . . 103
4.2 Study subjects . . . 104
4.3 Gene expression analysis . . . 104
4.4 Discussion . . . 112

5 Technical results **115**
5.1 Included patients and samples . . . 115
5.2 Sample distribution along the vertices of a multi-dimensional hypercube . . . 117
5.3 Validation of deconvolution approach: proof of concept data . . . 123

6 An unbiased view of all array data **127**
6.1 Data . . . 127
6.2 Quality control plots . . . 128
6.3 Sample relations . . . 132
6.4 Discussion . . . 138

7 Relation of IMPI-MA data to previous studies of blood transcriptomics in tuberculosis **139**
7.1 Data . . . 139

7.2 Study subjects . . . 140
7.3 Analysis . . . 140
7.4 Comparisons . . . 147

8 An analytic framework for heterogenous microarray data 155

8.1 The sample hypercube revisited: what kinds of questions relating to the three main hypotheses can we ask, and how can we answer these questions? . . . 155
8.2 Code description . . . 158
8.3 Computational output . . . 162
8.4 Results as Data: a new resource for generating and analysing HIV-TB related hypotheses . . . 164

III IMPI-MA: Results in depth 165

9 Question 1: Tuberculosis 167

9.1 Differential gene expression . . . 167
9.2 Array deconvolution and cell-specific differential gene expression . . . 173
9.3 Weighted gene co-expression network analysis (WGCNA) . . . 178

IV IMPI-MA: Results in breadth 187

10 Contrasts involving tuberculosis 189

10.1 Question 1: Tuberculosis . . . 189
10.2 Question 2: PTB and extrapulmonary TB . . . 197
10.3 Question 3: Latent TB . . . 204
10.4 Question 4: Haemodynamic phenotypes in TB-PC . . . 211
10.5 Pathway analysis . . . 217

11 Contrasts involving HIV infection status 229

11.1 Question 5: HIV (not active TB) . . . 229
11.2 Question 6: HIV (active TB) . . . 236

11.3 Pathway analysis 243

12 Contrasts involving the compartment **251**

12.1 Question 7: Blood and pericardial fluid in TB-PC 251

12.2 Pathway analysis 260

V Overall conclusions **271**

13 Conclusions and future work **273**

13.1 Summary of novel results 273

13.2 An approach to heterogeneous microarray data 274

13.3 Choice of methods 275

13.4 Results 275

13.5 Future work 277

VI Appendix **295**

A Code **297**

A.1 Code for functions used throughout 297

A.2 System setup 310

A.3 Data manager (general) 312

A.4 RT-PCR 317

A.5 Relation of IMPI-MA data to BERRY dataset 332

A.6 Overview of all data 355

A.7 Data manager for 7 questions 361

A.8 Analysis: IMPI-MA 366

List of Figures

1.1 **Clinical interactions of tuberculosis and HIV-1** . . . 33
1.2 **Levels of organisation in biological systems** . . . 42
1.3 **A systems approach is an iterative process** . . . 44

2.1 **Relationship between various IMPI cohorts** . . . 54
2.2 **The difference between the concepts *Contrast* and *Comparison*** . . . 59
2.3 **3D parameter space** . . . 60

3.1 **RT-PCR: calculation of ΔCT** . . . 62
3.2 **Clinical workflow for retrospective TB-PC cases** . . . 67
3.3 **Clinical workflow for retrospective controls** . . . 69
3.4 **Clinical workflow for prospective TB-PC cases** . . . 69
3.5 **Layering of components after Ficoll separation** . . . 72
3.6 **Workflow for data integration** . . . 76
3.7 **RNA preparation schema** . . . 79
3.8 **Workflow for RNA preparation** . . . 80
3.9 **Workflow for array hybridisation** . . . 80
3.10 **Approach for unification of cell type subsets** . . . 92

4.1 **Overall expression values** . . . 105
4.2 **Expression values by compartment** . . . 107
4.3 **Calibrated heatmap of all gene expression values** . . . 108
4.4 **Correlation matrices for 42 genes** . . . 109
4.5 **Calibrated heatmap of filtered gene expression values** . . . 109

4.6 Volcano plot of differentially expressed genes . . . 110
4.7 Calibrated heatmap of differentially expressed genes . . . 111
4.8 Principal component analysis . . . 112

5.1 Study subject recruitment . . . 116
5.2 RNA quality metrics . . . 117
5.3 Sample classes . . . 118
5.4 Three-dimensional sample phenotype space . . . 119
5.5 Extension of three-dimensional cube to six dimensions . . . 120
5.6 Six dimensional hypercube . . . 121
5.7 Three-dimensional phenotype space for LTBI . . . 121
5.8 Three-dimensional phenotype space for TB-site . . . 122
5.9 Three-dimensional phenotype space for HD phenotype . . . 122
5.10 Barplots of proportions in 11 samples . . . 124
5.11 Correlations of cell proportions . . . 125

6.1 PDF plots of all IMPI-MA arrays . . . 129
6.2 CDF plots of all IMPI-MA arrays . . . 130
6.3 Pairwise sample correlation . . . 131
6.4 MA plots for five arrays . . . 132
6.5 Box-and-whisker plot of all IMPI-MA arrays . . . 133
6.6 MDS: All IMPI-MA data . . . 135
6.7 MDS: IMPI-MA blood samples (HIV-TB) . . . 135
6.8 MDS: IMPI-MA blood samples (Sex/steroids) . . . 136
6.9 MDS: IMPI-MA pericardial fluid samples (HIV-TB) . . . 137

7.1 Box-and-whisker plots of the raw expression data for the three datasets . . . 142
7.2 Multidimensional scaling analysis . . . 143
7.3 Heatmaps of selected probes . . . 145
7.4 Multidimensional scaling analysis of selected probes . . . 146
7.5 Heatmaps of differentially expressed probes . . . 148

7.6 **Multidimensional scaling analysis of differentially expressed probes** 149
7.7 **Method validation** 150
7.8 **Data similarity** 151
7.9 **Probability of probe overlap** 151
7.10 **Overlap of all three datasets with the 393 probe list** 151
7.11 **Heatmaps of 61 overlapping probes applied to three datasets** 153

8.1 **Sample hypercube** 157
8.2 **Overall code structure for IMPI-MA main script** 159

9.1 **Data QC: not normalised (left) and normalised (right)** 168
9.2 **Volcano plot of differentially expressed probes** 170
9.3 **Heatmaps of significant and top 500 probes** 171
9.4 **PCA of differentially expressed probes** 172
9.5 **Overlap of differentially expressed probes for two different methods** 172
9.6 **Barplots of detected and PBMC types** 174
9.7 **Boxplots of detected and PBMC types** 176
9.8 **False discovery rate plots after cell-specific DE** 178
9.9 **Cell-type specific differential expression** 179
9.10 **WGCNA: Traits and modules** 180
9.11 **Module/ trait relationships** 182
9.12 **Top 3 GS/MM plots** 183
9.13 **Three significant modules** 184
9.14 **Weighted gene co-expression networks** 185

10.1 **Results for contrast *TB status* in contexts *HIV negative* and *HIV positive*** . . . 194
10.2 **Cell-type specific differential expression** 195
10.3 **Module clustering** 196
10.4 **Overlap of differentially expressed probes** 196
10.5 **Results for contrast *TB site* in contexts *HIV negative* and *HIV positive*** 201
10.6 **Cell-type specific differential expression** 202

10.7 **Module clustering** . . . 202
10.8 **Overlap of differentially expressed probes** . . . 203
10.9 **Results for contrast *LTBI* in contexts *HIV negative* and *HIV positive*** . . . 207
10.10 **Cell-type specific differential expression** . . . 209
10.11 **Module clustering** . . . 210
10.12 **Overlap of differentially expressed probes** . . . 210
10.13 **Results for contrast *Haemodynamic phenotype* in contexts *Blood* and *Fluid*** . . . 214
10.14 **Cell-type specific differential expression** . . . 215
10.15 **Module clustering** . . . 216
10.16 **Overlap of differentially expressed probes** . . . 216
10.17 **Overlap of differentially regulated pathways** . . . 221
10.18 **Significant pathways in tuberculosis (HIV-1 uninfected)** . . . 222
10.19 **Significant pathways in tuberculosis (HIV-1 infected)** . . . 223
10.20 **Toll-like receptor pathway** . . . 225
10.21 **Antigen processing and presentation pathway** . . . 226
10.22 **NK cell cytotoxicity pathway (HIV-1 uninfected)** . . . 227

11.1 **Results for contrast *HIV status* in contexts *Healthy* and *LTBI*** . . . 233
11.2 **Cell-type specific differential expression** . . . 234
11.3 **Module clustering** . . . 235
11.4 **Overlap of differentially expressed probes** . . . 235
11.5 **Results for contrast *HIV status* in contexts *TB-PC Blood* and *TB-PC Fluid*** . . . 240
11.6 **Cell-type specific differential expression** . . . 241
11.7 **Module clustering** . . . 242
11.8 **Overlap of differentially expressed probes** . . . 242
11.9 **Significant pathways in HIV** . . . 246
11.10 **Overlap of differentially regulated pathways** . . . 247
11.11 **Antigen processing and presentation pathway** . . . 249
11.12 **RIG-I-like receptor signaling pathway** . . . 250

12.1 **Results for contrast *Compartment* in contexts *TB-PC HIV neg* and *TB-PC HIV pos*** . . . 256
12.2 **Barplots and boxplots of matched blood and pericardial fluid samples** . . . 257
12.3 **Cell-type specific differential expression** . . . 258
12.4 **Module clustering** . . . 259
12.5 **Overlap of differentially expressed probes** . . . 259
12.6 **Overlap of differentially regulated pathways** . . . 266
12.7 **Antigen processing and presentation pathway** . . . 268
12.8 **Natural killer cell mediated cytotoxicity** . . . 269

List of Tables

3.1 **42 genes selected for RT-PCR analysis** . . . 63
3.2 **Definitions of terms** . . . 66
3.3 **Sources of clinical phenotype data used in IMPI-MA** . . . 74
3.4 **Levels of data complexity** . . . 81
3.5 **Methods for selecting differentially expressed genes** . . . 85

4.1 **Clinical characteristics of study participants** . . . 104
4.2 **Gene-by-gene comparison of expression between compartments** . . . 106
4.3 **Differentially expressed genes** . . . 110

5.1 **Reasons for exclusion of RNA samples** . . . 116

7.1 **Table of clinical characteristics across datasets and contrasts** . . . 141
7.2 **Effect of non-specific filtering** . . . 141
7.3 **Results of differential expression analysis** . . . 147

8.1 **Questions** . . . 156

9.1 **Top 30 probes as selected by *limma*** . . . 169
9.2 **Significance of differences in cell proportions in active TB vs not active TB** . . . 175
9.3 **Significance of modules for contrast "active TB vs no TB"** . . . 181

10.1 **Clinical characteristics: active TB vs not active TB** . . . 191
10.2 **Overall results: active TB vs not active TB** . . . 192
10.3 **Clinical characteristics: PTB vs TB-PC** . . . 198
10.4 **Overall results: PTB vs TB-PC** . . . 199

10.5 **Clinical characteristics: Healthy vs LTBI** . . . 205
10.6 **Overall results: Healthy vs LTBI** . . . 206
10.7 **Clinical characteristics: Effusive vs Effusive-constrictive pericarditis** . . . 211
10.8 **Overall results: Effusive vs Effusive-constrictive pericarditis** . . . 212
10.9 **Top pathways in TB (HIV-1 uninfected)** . . . 218
10.10 **Upregulated pathways in TB (HIV-1 infected)** . . . 219
10.11 **Downregulated pathways in TB (HIV-1 infected)** . . . 220

11.1 **Clinical characteristics: HIV positive vs HIV negative (not active TB)** . . . 230
11.2 **Overall results: HIV positive vs HIV negative (not active TB)** . . . 231
11.3 **Clinical characteristics: HIV positive vs HIV negative (active TB)** . . . 237
11.4 **Overall results: HIV positive vs HIV negative (active TB)** . . . 238
11.5 **Upregulated pathways in HIV (not active TB)** . . . 243
11.6 **Downregulated pathways in HIV (not active TB)** . . . 244
11.7 **Upregulated pathways in HIV (active TB)** . . . 245
11.8 **Downregulated pathways in HIV (active TB)** . . . 245

12.1 **Clinical characteristics: blood vs pericardial fluid** . . . 252
12.2 **Overall results: blood vs pericardial fluid** . . . 254
12.3 **Pathways significantly upregulated in pericardial fluid (HIV-1 uninfected)** . . . 261
12.4 **Pathways significantly downregulated in pericardial fluid (HIV-1 uninfected)** 262
12.5 **Pathways significantly upregulated in pericardial fluid (HIV-1 infected)** . . . 263
12.6 **Pathways significantly downregulated in pericardial fluid (HIV-1 infected)** . . . 264

Listings

A.1 heatmapad.R . . . 297
A.2 superHeatmap.R . . . 302
A.3 superHeatmap2.R . . . 304
A.4 plotSampleRelationsAD.R . . . 306
A.5 plotSampleRelations3DAD.R . . . 308
A.6 SystemSetup.R . . . 310
A.7 Data-manager.R . . . 312
A.8 RT-PCR code . . . 317
A.9 RelationIMPI-MA to TB-AOG . . . 332
A.10 Overview of all data . . . 355
A.11 Data-manager-Q.R . . . 361
A.12 SevenQuestions . . . 366

Terminology and abbreviations

AFB	Smear for acid-fast bacilli (either Ziehl Neelsen or immunofluorescence)
BERRY	Data used in Berry, et al (Nature, 2010)
BSL-3	Biosafety Level 3
CDF	Cumulative density function
CFF	Cell-free pericardial fluid
FDR	False discovery rate
HIV+	HIV-1 infected (HIV positive)
HIV-	HIV-1 uninfected (HIV negative)
HIV-TB	Co-infection with HIV-1 and *Mycobaterium tuberculosis*
IMPI-MA	Investigation into the management of tuberculous pericardits: microarray sub-study
KEGG	Kyoto Encyclopedia of Genes and Genomes
LTBI	Latent tuberculosis infection
NIMR	National Institute for Medical Research
PBMCs	Peripheral blood mononuclear cells
PBS	Phosphate-buffered saline solution
PCF	Pericardial fluid

PDF	Probability density function
PFC	Pericardial fluid cells
PTB	Pulmonary tuberculosis
RPMI+10%FCS	Roswell Park Memorial Institute medium (version 1640) containing 10% heat-inactivated fetal calf serum
RPMI-1640	Roswell Park Memorial Institute medium (version 1640)
TB	Tuberculosis
TB-PC	Tuberculous pericarditis
TOM	Topological overlap matrix
WGCNA	Weighed gene network co-expression analysis

Part I

Introduction and methods

1 HIV/TB and studies at the site of disease

Chapter summary

In this chapter I describe in detail the clinical, cellular and molecular characteristics of co-infection with *M. tuberculosis* and HIV-1 (HIV-TB). I go on to describe systems approaches in general and applied to tuberculosis, and finally argue for system-level studies in HIV-TB.

The text of this Chapter is taken verbatim from a review paper submitted to Pathogens and Disease on 17 April 2013 and accepted for publication on 20 June 2013 [1]. Authors of this paper are Armin Deffur (first author), Nicola Mulder and Robert J Wilkinson (senior author); the co-authors provided input in line with their roles as supervisors. The rationale for including this publication in the thesis is that it provides a comprehensive literature review of the problem of HIV-1 and *Mycobacterium tuberculosis* co-infection. Please note that British English spelling rules have been applied to the text where relevant.

1.1 Introduction/outline

Co-infection by HIV-1 and *Mycobacterium tuberculosis* is of great concern worldwide. In this review I present a concise overview of clinical and immunological features of HIV-TB, review experimental systems in which the biology of this co-infection is studied and finally discuss how a systems approach may be employed in advancing understanding of these conditions.

1.2 Clinical features and importance of HIV-TB

About a third of the human population are estimated to be infected with *M. tuberculosis*. In the majority of cases the infection is clinically inapparent, with only approximately 10% of individuals at risk of developing overt clinical tuberculosis in their lifetime.

This indicates a fine balance between effective host defences (resulting in bacterial containment or clearance) and substantive infection that results in significant pathology, potentially killing the host early in the course of infection. Either scenario is clearly not beneficial for the reproductive success of *M. tuberculosis*, and a delicate balance has evolved that maintains the host-pathogen relationship over long time periods. Even in the case of a shift towards active infection with concurrent pulmonary pathology, the resulting disease (pulmonary tuberculosis) can follow a chronic course with 37% 5 year survival if left untreated [2]. The recent arrival of HIV-1 as a human pathogen has shifted the balance towards a dramatic increase in clinical disease and increased mortality associated with tuberculosis in settings where HIV-1 is highly prevalent. This has resulted in a deadly syndemic [3], where the consequences of co-infection manifest variability in clinical and pathologic features and a shift towards worse outcome. The pathobiology of this clinically important co-infection is incompletely understood. While multiple risk factors have substantial impact on the worldwide tuberculosis epidemic, HIV-1 infection is an important driver of tuberculosis. 13% of worldwide incident cases of tuberculosis are HIV-1 co-infected, and HIV-1 remains the most important risk factor in many high-burden settings [3]. HIV-TB accounts for a disproportionate share of tuberculosis-related mortality. It is therefore instructive to review in outline what is known about the interaction of HIV-1, *M. tuberculosis* and the human immune system at the individual level. Population-level interactions will not be reviewed here.

1.2.1 Effect of HIV-1 on tuberculosis

HIV-1 co-infection increases risk for TB susceptibility to primary or re-infection. While in general HIV-TB patients are less frequently sputum positive, their greater numbers in high-burden settings for both tuberculosis and HIV-1 infection, suggests they may contribute substantially to transmission. A study in Harare found that HIV-1 infected individuals had high levels of incident TB, while HIV-1 uninfected individuals had higher levels of prevalent smear-positive tuberculosis, suggesting

that this subpopulation still drives transmission of *M. tuberculosis*, while HIV-1-infected individuals account for the majority of incident cases [4]. Of all opportunistic infections with associated with HIV-1, the risk for developing clinical tuberculosis is increased shortly after HIV-1 infection has taken place, well before the total CD4 count drops to levels below 500 cells per μl [5]. This raises the question whether HIV-1 causes an early tuberculosis-specific defect in host immunity. HIV-1 also increases the risk of progressing from one stage of tuberculosis to the next [6], including increased risk of primary disease following exposure [7], reactivation of latent (clinically quiescent) lesions [8, 2]and reinfection with *M. tuberculosis* following exposure [9, 10, 3].

Tuberculosis disease phenotypes are also altered by HIV-1 co-infection. While cavitatory pulmonary tuberculosis is the most common clinical manifestation in HIV-1-uninfected individuals, atypical presentations of pulmonary disease are much more prevalent in HIV-1 infected individuals [11, 12, 3]. This tendency to atypical pulmonary disease is linked to CD4 count, inasmuch as individuals with preserved CD4 counts present with upper lobe cavitation much like their HIV-1 uninfected counterparts, whereas low CD4 counts are linked to atypical infiltrates (middle and lower lobe infiltrates, interstitial nodules), mediastinal lymph node involvement and even normal chest radiographs [13]. While chest radiographs are unhelpful in detecting active tuberculosis in HIV-1 infected individuals in many instances, so is the addition of Interferon-gamma release assays to screening strategies aimed at identifying smear-negative tuberculosis [14]. In addition to pulmonary disease, which remains the most important presentation in HIV-1 co-infected individuals, extrapulmonary forms of tuberculosis are more common in this group compared to HIV-1 uninfected individuals [15, 5]. While extrapulmonary tuberculosis usually presents with a single site in the absence of HIV-1 infection, HIV-1-co-infected individuals often present with concurrent pulmonary and extrapulmonary disease.

In a prospective study of extrapulmonary tuberculosis, which consisted of HIV-1 infected and uninfected individuals, the most common extrapulmonary sites were lymph nodes (43%), pleura (23%), central nervous system (8%), musculo-skeletal system (7%), genitourinary tract (5%), and gastrointestinal tract (2%) [6, 16] with other sites making up the remainder (Data was not stratified by HIV-status in original study). Finally, HIV-1-co-infection increases tuberculosis-associated mortality[7, 17].

1.3 Clinical phenotypes and diagnosis: HIV-TB is a heterogeneous syndrome

Tuberculosis infection in humans is classically divided into two distinct phenotypes[18]. In this paradigm, latent tuberculosis infection results from primary infection of the lung that is contained but not eradicated by a host adaptive immune response. This response is characterised by granuloma formation, intracellular (mainly within macrophages) location of mycobacteria, which do not replicate and may be metabolically adapted to persistence, and the absence of any symptoms suggestive of tuberculosis. The other phenotype is active pulmonary tuberculosis, (usually ascribed to reactivation of a previously dormant lesion). This century-old description is being increasingly questioned, and current thinking favours the concept of a spectrum of disease phenotypes [19], ranging from cleared infection without or with immune sensitisation, inactive lesions containing mycobacteria, active lesions with bacterial replication but without symptoms to full-blown active disease (See Figure 1.1).

HIV-1 co-infection complicates this picture by skewing clinical and pathological phenotypes towards more advanced disease, significantly contributing to the heterogeneity of clinical presentations of tuberculosis in the context of HIV-1 co-infection. In order to dissect this complexity, more detailed phenotypes of human disease may be generated by using novel functional imaging techniques [20, 21, 19].

Primary disease following recent tuberculosis exposure is also more common in HIV-1 co-infection, as is an increased risk for reactivation of previously quiescent lesions as well as re-infection of persons already infected with *Mycobacterium tuberculosis*. The question of whether prior or current tuberculosis elicits an immune response that can protect from infection is controversial. While a 2005 review [22] found that evidence for protective immunity (against re-infection) induced by latent tuberculosis was not convincing, a recent systematic review [23] showed that individuals with latent tuberculosis had a 79% lower risk of progressive tuberculosis after re-infection than unsensitised individuals.

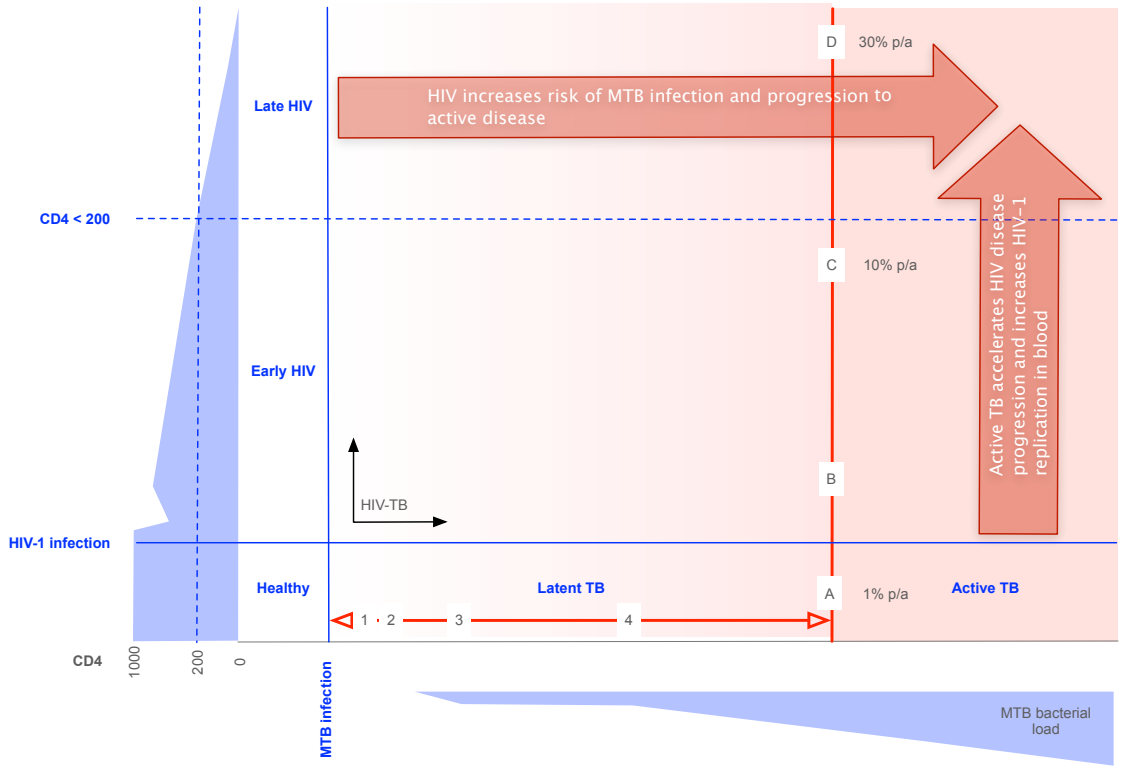


Figure 1.1: **Clinical interactions of tuberculosis and HIV-1.** HIV-1 and tuberculosis interact with each other. The x-axis represents stages of tuberculosis, from infection through to active disease, while the y-axis represents stages of HIV-1 infection. *M. tuberculosis* bacterial burden and CD4 count are shown in blue along the respective axes. The spectrum of "latent" TB is represented as follows: (1) Infection eliminated without priming antigen-specific T-cells. (2) Infection eliminated in association with T-cell priming. (3) Infection contained with some bacteria persisting in a non-replicating form. (4) Bacterial replication maintained at subclinical level by the immune system. Clinical disease (pulmonary and extrapulmonary tuberculosis) occurs in a subset of individuals who are latently infected or who develop primary tuberculosis directly following infection or re-infection. The annual risk is represented as follows: (A) HIV-1 uninfected: approximately 10% lifetime risk, or about 1% per annum (p.a.). (B) Shortly after HIV-1 infection, and prior to substantial CD4 T-cell depletion, the risk for active tuberculosis increases. (C) During the early stages of HIV-1 infection, this risk rises to approximately 10% p.a. (D) In late-stage HIV-1 infection, the risk for active tuberculosis increases to 30% per annum. The effects of HIV-1 on tuberculosis, and of tuberculosis on HIV-1 disease are shown by the red arrows.

1.4 Immune reconstitution after ART

Following initiation of highly active antiretroviral therapy (ART) in patients with HIV-1 infection, tuberculosis incidence is reduced as protection from tuberculosis is restored with recovery of CD4 count. This reduction in incidence is only partial, however, as tuberculosis incidence still remains higher after ART and CD4 restoration compared to HIV-1 uninfected individuals [24]. As reconstitution of a protective immune response against many antigens occurs, individuals may also develop worsening symptoms and signs of opportunistic infections due to a vigorous response against antigens present in the host resulting from immune activation that may be modulated by corticosteroids [25]. This is termed tuberculosis immune reconstitution inflammatory syndrome (TB-IRIS). TB-IRIS appears to be driven by increased recognition of *M. tuberculosis* antigens present shortly after ART initiation; this may be associated with conversion of a negative TST to a strongly positive one.

Pathogenic mechanisms that have been considered in TB-IRIS include purified protein derivative (PPD)-specific expansion of Th1 responses [26] and defective restoration of regulatory T cell function. However, Th1-specific expansions may also occur in individuals who do not develop symptoms and signs of TB-IRIS [27, 25], and numbers of CD4+ Foxp3+ T cells in TB-IRIS cases are similar in controls[28]. More recent work has focused on the interaction of innate and adaptive responses, with evidence of myeloid activation [29, 30], cytolytic action of NK cells [31] as well as killer immunoglobulin receptor (KIR)-negative $\gamma\delta$-T cells [32] all contributing to TB-IRIS.

In contrast to the pathologic responses associated with TB-IRIS, functional, non-pathologic immune restoration is responsible for the enhanced protection from active tuberculosis seen following start of ART. This response is associated with an increase in absolute CD4 T cell count, decline in HIV-1 viral load, an absolute increase in effector function but a proportional decrease in the ratio of effector to central memory CD4 T cells due to an even greater expansion of the central memory pool, which provides the strongest correlate of ART-mediated immune restoration [33]. Another study found that the extent of immune restoration following ART initiation depended on the baseline (nadir) CD4 T cell count. Only at nadir CD4 counts greater than 350 cells/ μl did the immune restoration result in CD4 T cell populations similar to those in HIV-1 uninfected individuals, suggesting an immunological benefit of starting ART at that threshold [34].

1.5 Disease mechanisms in co-infection

1.5.1 Immune mechanisms in tuberculosis

Immune control of *M. tuberculosis*, an intracellular pathogen relies on both innate and adaptive cell-mediated immunity [35]. Although a vigorous humoral response to multiple mycobacterial antigens is produced, the antibodies appear to be largely ineffective in controlling the disease due to the mainly intracellular localisation of the bacterium; there is some debate whether opsonising antibody may be helpful early in infection. The response therefore requires CD4 and CD8 T cells that respond to *M. tuberculosis* antigens presented by either MHC class I, MHC class II or CD1 molecules on the surface of antigen presenting cells (macrophages and dendritic cells), a Th1 cytokine milieu (IFN-gamma, IL-12, TNF) and appropriate regulatory responses that limit the extent of the inflammatory response. Even in active tuberculosis, there is a delicate balance between pro- and anti-inflammatory responses, that aim to contain the infection while limiting collateral damage. This results in the often very protracted course of pulmonary tuberculosis, recognised as the syndrome of "consumption" prior to the era of antimycobacterial chemotherapy. Co-infection with HIV-1 has a dramatic impact on this balance.

1.5.2 Effect of HIV-1 on tuberculosis immunity

The most obvious effect on tuberculosis immunity by HIV-1 is the severe depletion of the CD4 T cell population, both in blood and in peripheral lymphoid tissues [36, 37]. In addition, HIV-1 induces CD4 T cell dysfunction as evidenced by reduced IL-2 or IFN-gamma production[38, 39, 40]. Early selective depletion of *M. tuberculosis* antigen-restricted CD4 cells was demonstrated in 2008, and likely explains the early increase in risk of tuberculosis following HIV-1 infection [41, 42]. It is hypothesised that HIV-1 may indirectly cause loss of *M. tuberculosis*-specific CD8 T-cells through loss of *M. tuberculosis*-specific CD4 T cells, though this has not yet been experimentally demonstrated in humans. CD8 T cells probably play an important role in containing tuberculosis infection. This is view is based on data from *in vitro* work [43], mouse models [44], non-human primate models [45], and cells isolated from individuals with tuberculosis following anti-TNF therapy [46].

1.5.3 Effect of tuberculosis on HIV-1: viral replication

Infection with *Mycobacterium tuberculosis*, and more directly the immune response directed against it, result in enhanced intracellular HIV-1 replication in blood [47] and at sites of tuberculosis disease, including the lung [48] and pericardial space in the case of tuberculous pericarditis [49]. Multiple mechanisms have been implicated, including cytokine- and chemokine-mediated mechanisms[50, 51], loss of an inhibitory transcription factor C/EBP beta[52], activation of (NF)-κB and signalling involving positive transcription elongation factor (P-TEFβ) [53]. These mechanisms cause activation of transcription of HIV-1, resulting in enhanced viral replication, spread to uninfected cells and their subsequent depletion.

1.5.4 Effect of tuberculosis on HIV-1: clinical disease progression

Active tuberculosis is associated with more rapid progression to severe CD4 T cell depletion and higher risk for opportunistic infections resulting in late stage HIV-1 infection and the development of AIDS [54, 55].

1.5.5 The role of host genetic variation

There is moderate evidence of human genetic susceptibility to tuberculosis [56]. For example, using a genome-wide screening approach in *M. tuberculosis*-infected vs. non-infected dendritic cells, Barreiro et al found a number of expression quantitative trait loci (eQTL) that associate with tuberculosis in this system [57], and imputation of 1000 Genomes Project data [58] into a Ghanaian genome-wide dataset (tuberculosis cases and healthy controls) revealed a resistance locus at 11p13 downstream of WT1 (Wilms tumor 1) [59]. Thus, host genetics introduces an additional factor to consider when discussing the interaction between HIV-1 and tuberculosis. Recently, susceptibility to tuberculosis in HIV-1 infected individuals was associated with a CARD8 genetic variant [60], and a genotype resulting in high IL-10 production also appeared to predispose HIV-1 infected individuals to TB infection [61]. It should be noted however, that studies that infer genetic risk for tuberculosis are often inconsistent in their results, and study design is probably responsible for this [62]. The definition of tuberculosis phenotype often varies across such studies, making them hard to compare directly and to generalise their results. This caveat applies equally to studies of

tuberculosis susceptibility in HIV-1 infected individuals.

1.6 Experimental systems

1.6.1 Isolated model systems

Experimental systems consisting of single cell types (e.g macrophages) manipulated *in vitro* are useful for delineating actions of individual mediators (cytokines, chemokines), intracellular processes (signalling cascades, autophagy) and processes that are ultimately reductionist. While yielding invaluable data on the "parts list" of the immune response to tuberculosis, such systems by definition lose higher-order information about the overall orchestration and coordination of the immune response.

1.6.2 Animal models

Animal models of tuberculosis can be very useful in understanding higher-level immune phenomena that are lost in simple systems, but care must be taken when extrapolating these phenomena to human immunobiology without experimental evidence.

Early major contributions of the mouse model to understanding human tuberculosis are insights into the key importance of IFN-γ, TNF-α and CD4 T-cells in containing tuberculosis. The mouse model is utilised to study the effect of *M. tuberculosis* genomic variation (including induced mutants), immune response, the effect of drugs and vaccines [63].

An argument against the use of mouse models is that they cannot effectively recapitulate HIV-1 co-infection in tuberculosis. Considerable effort has gone to develop mouse models of HIV-1 infection [64, 65]. Basic models aim to mimic HIV-1 infection by CD4 T-cell depletion. While CD4 depletion is clearly important in HIV-1 pathogenesis, HIV-1 has additional effects on the immune system that are not captured in this model (apoptosis, disruption of lymph node architecture, depletion of *M. tuberculosis*-specific T cells and effects on macrophage function). Therefore, more realistic models have been developed, including Nef-transgenic mice [66, 67, 68] and humanised bone marrow-liver-thymus (BLT) mice [69, 70, 71] that support the entire life cycle of HIV-1 and thus may become valuable tools in understanding aspects of tuberculosis and HIV-1 co-infection.

In contrast to their indisputable utility in discovering disease mechanisms in tuberculosis [72], use of mouse models at the system level (by the use of unbiased genomic screens) may also lead to incorrect conclusions if great care is not taken when mapping mouse data onto human orthologs. In a recent study, transcriptomic responses to three broad inflammatory stimuli were studied in humans and mice using a whole-blood transcriptomic approach [73]. In humans, the transcriptomic responses were correlated with each other, but the corresponding mouse models of the inflammatory conditions (burns, trauma, endotoxinaemia) were both uncorrelated with their human counterparts, and with each other. This strongly suggests that overall control in the mouse immune system may be wired differently than in humans, and that large-scale organisational features are not conserved across this particular species barrier. This finding argues strongly to restrict systems-level investigations into tuberculosis and HIV-1 co-infection to models other than mice, such as human (clinical) and non-human primate models.

In contrast to mouse models, other animal models recapitulate more features of human tuberculosis, e.g. guinea pigs and rabbits both produce caseous granulomas, a hallmark of human disease, but are limited by the availability of immunological reagents required to interrogate the models. The non-human primate model is the closest approximation of human tuberculosis [74, 75]. Longitudinal samples including blood, broncho-alveolar lavage fluid and lymph node biopsy tissue are available during the infection stage of the model, and all tissues are available at necropsy. As non-human primates can be infected with simian retroviruses that cause an immunodeficiency syndrome in these animals, use of this model may extend to study the non-human primate equivalent of HIV-TB [65]. Given these features, this model system is able to generate data that is essentially impossible to generate from other animal models or human clinical disease models.

Despite their utility, all animal models are limited when performing system-level studies that aim to understand precise mechanisms of tuberculosis and HIV-1 co-infection in humans. Given these limitations it is imperative that an integrated, systems-level model of tuberculosis and HIV-1 co-infection be developed for humans in an unbiased way using systems approaches.

1.6.3 Difficulty of *in vivo* work in humans

In vivo studies of human tuberculosis are difficult, as sampling of disease sites cannot be carried out in the same way as in animal models. Post-mortem studies of human tuberculosis are available, but relatively limited, and the ability to perform experimental challenges are very limited.

The bulk of our existing knowledge comes from technologies that may suffer from low spatial resolution (e.g. chest radiographs may be normal in subclinical pulmonary disease) that may fail to resolve phenotypes along the latency spectrum, and obscure the inherent heterogeneity of human tuberculosis.

Our understanding of this low-resolution data has led to classical descriptions of tuberculosis pathogenesis involving distinct, polarised phenotypes: Infection with *Mycobacterium tuberculosis* leading to either latent tuberculosis or primary progressive disease, and reactivation (usually in the lung apices) of latent infection which leads to full-blown clinical tuberculosis.

Compounding the problem is the fact that the early stages of tuberculosis are hard to study. It is likely that very early events following exposure to aerosols containing *M. tuberculosis* are key in determining the outcome of the exposure. Hypothetical outcomes are threefold:

1. Clearance of inhaled *M. tuberculosis* by innate immune responses (possibly mediated by neutrophils alone or in combination with other innate cells) without presentation of cognate antigen to adaptive immune cells, therefore not resulting in any acquired response.

2. Failure of the innate immune system to clear the infection, resulting in cells of the adaptive immune system becoming actively involved, with antigen presentation on dendritic cells and macrophages to T-lymphocytes and subsequent downstream events. At this stage the infection may be cleared or contained within granulomas. IGRA and/ or TST responses are used as indicators of this immune sensitisation to *M. tuberculosis*, but cannot differentiate between cleared infection with a lasting memory response, a contained infection that may yet reactivate, and subclinical disease.

3. The infection may overwhelm innate and adaptive responses and result in primary progressive disease.

1.6.4 Limitations of reductionist approaches

Current reductionist approaches have defined a multitude of genes, proteins, cellular mechanisms and cell populations that are important in containing mycobacterial infection, account for immune deficits caused by HIV-1 infection and describe the response to vaccines. However, these molecular and cellular-level mechanisms interact to produce high-level phenomena, such as protective immune responses. At this stage, it is unknown what truly constitutes an immune response that protects from infection with *M. tuberculosis* or from progressive disease following infection. The simple single cell models and even animal models do not recapitulate human HIV-TB co-infection completely. Simple models interrogate only a few dimensions of a very high-dimensional parameter space that describes the complex system fully. As a result, a large portion of this parameter space remains unexplored. HIV-1 co-infection adds to this complexity by introducing one or more orthogonal dimensions to the parameter space. An alternative is a systems approach to HIV-TB co-infection that yields models that are able to predict emergent phenomena.

1.6.5 Evolutionary perspective

The association of *Mycobacterium tuberculosis* with human hosts is not a recent phenomenon, but rather a long-standing one, with co-evolution of host and pathogen [76]. *M. tuberculosis* is found wherever human populations are found, and at least seven distinct lineages have been identified [77]. In addition, humans have evolved while exposed to multiple retroviruses, which remain as genomic remnants in the form of HERVs (human endogenous retroviruses) [78]. Also, HIV-1 infection disables key components of the adaptive immune response to tuberculosis (some specific for *M. tuberculosis* responses) while leaving innate mechanisms more intact. HIV-1 and *M. tuberculosis* may affect each other's evolution. An increase in HIV-1 heterogeneity may be associated with a TB-mediated increase in HIV-1 replication discussed above, suggesting the possibility that HIV-1 and *M. tuberculosis* co-infection may alter the evolutionary dynamics of HIV-1[79]. Conversely, an increase in multidrug-resistance conferring mutations in *M. tuberculosis* has sometimes been noted in HIV-1 co-infected individuals, suggesting that HIV-1 may directly or indirectly influence *M. tuberculosis* diversity [79].

Worldwide distribution of different lineages of *M. tuberculosis* mirrors that of geographically

defined human population groups, which suggest that the specific strains are adapted to their (geographically) local population. This sympatric relationship is disrupted following HIV-1 infection, in which a much higher likelihood of infection with allopatric strains was seen compared to HIV-1 uninfected individuals [80].

1.7 Systems approaches

1.7.1 Introduction to systems

A "system" can be loosely defined as a collection of entities that collectively exhibit particular behaviour. A hallmark of complex systems is that this behaviour is hard to predict when given information about the constituent parts. Systems are organised in levels of organisational complexity. A defining feature of systems is the concept of interaction: any particular level of organisation may be viewed as a set that has members consisting of specific biological entities that interact with each other. Usually, the set of interactions between the various members can be described using network terminology. An intuitive example is the collection of signalling pathways within a specific cell type that consists of interactions between ligands, receptors, downstream linker molecules and ultimately effector molecules (e.g. transcription factors) that alter gene expression. The set of interactions between entities of one level of organisation then give rise to the next level of organisation, a set that has more complex members (e.g. protein complexes, double-layer lipid membranes), which again interact with each other to form a higher-order system. This process is repeated from the subatomic scale to the population level. Figure 1.2 shows the different levels of organisation, components of those levels and examples of emergent properties resulting from members within that level.

Systems biology is an emerging discipline that aims to describe complex systems by studying the interactions within that system, and by modelling these interactions of known system constituents discovering emergent properties of the system that could not be predicted *a priori*. Using this approach of data-driven modelling, predictions about higher-order system behaviour can be made, and importantly these predictions may be tested in the system of interest.

Level	Components	Emergent phenomena
Population	Human populations	Transmission dynamics of M. tb
Multicellular Organism	Human	Disease state: Latent tuberculosis, active tuberculosis
Organ system	Respiratory system, immune system	Respiratory failure, cough, weight loss
Organ	Lung, lymph nodes, blood	Macroscopic pathology: caseating necrosis, cavitation, exudative effusions
Tissue	Respiratory epithelium, alveolar tissue, lymphoid tissue, recruited cells	Granuloma formation
Cell	*Mycobacterium tuberculosis* Macrophage, dendritic cell, CD4 T cell	*Mycobacterium tuberculosis*: growth, non-replicative state Human: immune cell activation, apoptosis
Organelle	Cell membranes, phagosomes, lysosomes	Phagocytosis, phagosome-lysosome fusion, phagosomal arrest, autophagy
Macromolecule	DNA, proteins (receptors, signal transduction) phospholipid assemblies	Signalling cascades
Molecule	Cytokines, chemokines, metabolites	Antigen recognition sites, ligand-receptor binding, metabolic networks
Atom	Individual atoms	Chemistry, biochemistry

Figure 1.2: **Levels of organisation in biological systems.** At each level, entities of the same class interact in complex interaction networks, and emergent properties of these entities and their interactions define the next level up. The first column lists the hierarchical levels of organisation of biological systems. Components that are studied at each particular level are shown in the second column (these components do not form an exhaustive list). In the third column, examples of emergent phenomena that result from interactions between components are listed.

1.7.2 Systems approaches

Systems approaches should be hypothesis driven, global/ unbiased and multi-scale. They should iterate over cycles of data collection, hypothesis formulation, model construction and model testing [81] (Figure 1.3). Data collection requires unbiased sampling of the system and parallel measurement of high numbers of parameters at multiple levels of organisation. This is facilitated by the advent of various high-throughput biology technologies that measure whole genome gene expression (DNA sequencing, cDNA/cRNA microarrays, RNAseq), protein levels or presence (mass spectrometry), small molecule metabolites and other large-scale phenomena. Such datasets may then be incorporated into models of gene expression, metabolic pathways and other models. Important in this context is the concept of perturbation, where the experimental system is subjected to specific changes, and the broad-based measurements mentioned above repeated. This data may then be used to test hypotheses about emergent properties of the system formulated using the model.

1.7.3 Systems approaches to tuberculosis

An important contribution of the systems-level understanding of tuberculosis was made in 2010[82]. In this study, the authors collected whole blood for RNA extraction from a cohort of individuals with either active pulmonary tuberculosis or not active pulmonary tuberculosis (comprising latent tuberculosis and individuals without evidence of immune sensitisation to *M. tuberculosis* antigens). Extracted and processed samples were hybridised to Illumina Human HT12v3 gene expression microarrays. The resulting data was first subjected to unsupervised analysis followed by statistical testing for significant differential transcript abundance. This approach [83, 84] yielded an interferon-inducible, neutrophil driven blood transcriptional signature consisting of 393 transcripts, shedding light on the potential role of neutrophils in active tuberculosis. Type-1 interferon-inducible genes were shown to be highly expressed in neutrophils. In subsequent work [85, 86] core features of this signature were independently replicated. This has led to new thinking regarding the role of the innate immune system in the response to tuberculosis, and its interaction with the adaptive response. This signature is being evaluated for diagnostic potential of early identification of active tuberculosis, and has also been shown to revert towards the inactive signature following antimycobacterial therapy [87]. An additional contribution of this paper was the finding that some

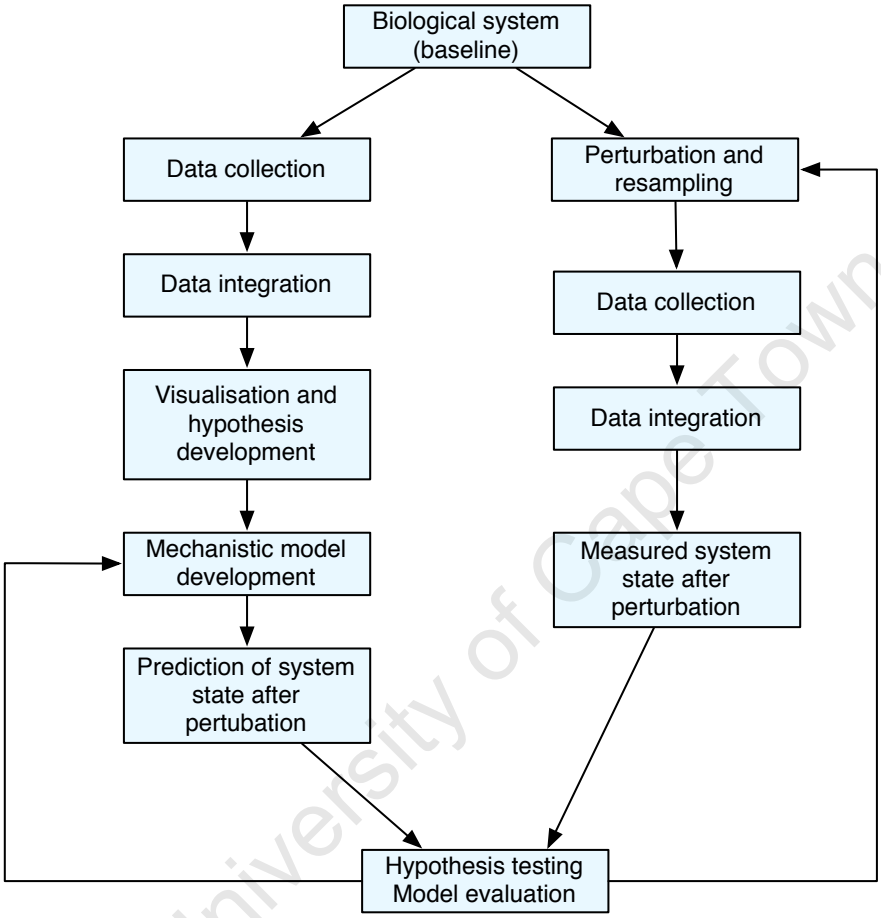


Figure 1.3: **A systems approach is an iterative process.** For each system under study, experimental data collection, data interpretation, modelling, prediction and further experimentation follow each other. The key goal is correct prediction of system states following perturbation of such systems using computational models.

individuals who had the clinical phenotype of latent tuberculosis assigned based on the absence of symptoms and evidence of prior immune sensitisation to *M. tuberculosis* clustered together with cases of active tuberculosis, suggesting that their correct phenotype was subclinical active tuberculosis [82], supporting the idea that tuberculosis should be viewed as a spectrum. As this interpretation lacks experimental support, alternative interpretations of the data should be considered. For instance, the misclassified individuals may have another subclinical condition that produces a transcriptional response that is similar to that associated with active tuberculosis. In this context, it is worth noting that the 86-transcript signature described by Berry et al also distinguishes individuals with melioidosis from healthy controls [88], suggesting that this signature may not in fact be specific for active tuberculosis. The paper concludes that type 1 and type 2 interferon-mediated responses dominate the host response to both infections, and that transcriptional signatures based on interferon signalling cannot distinguish between acute melioidosis and tuberculosis.

In developing systems approaches to tuberculosis, novel mathematical methods need to be developed and implemented such as scoring methods for protein interactions [89], investigation of gene co-expression patterns in *M. tuberculosis* using microarray data [90] and modelling latent tuberculosis using time-course microarray data [91]. Another approach to investigating tuberculosis has been developed by Kirschner et al. Here a systems biology approach [92, 93] is used to integrate experimental data (e.g. flow cytometry, enzyme-linked immunosorbent spot (ELISPOT), Luminex, immunohistochemistry, *in situ* hybridisation (ISH), *in vivo* and *in vitro* cytotoxicity assays, gene knockout (deletion) and cell and cytokine depletions, cytotoxicity assays, 2-photon microscopy) and computational models to study emergent properties of a system consisting of lung and draining lymph nodes [94] down to the cellular level, and find that many aspects of granuloma formation are recapitulated in this model [95]. A view of intracellular host factors important for survival of *M. tuberculosis* has been developed based on genome-wide siRNA screens, which yielded a dense interaction network of 275 molecules [96]. A subset of 74 host proteins was shown to be invariably required for survival of *M. tuberculosis*; over half of these act through regulation of autophagy which implies that modulation of autophagy forms a basis for intracellular survival.

An EU-funded consortium (SystemTB) aims to combine experimentally-driven model development and model-driven experimentation to improve the understanding of the biology of *M. tuberculosis* and its interactions with host cells and mediators by elucidating its interaction networks by

various high-throughput screens performed under multiple conditions involving biologically relevant perturbations (chemical and physical stresses, knock-out or knock-down mutants [97]. In the USA, one of four NIH-funded Systems Biology of Infection centres based at the Broad institute has established a collaborative network that will perform systematic profiling using high-throughput screens during both *in vitro* and *in vivo* growth, integrating these data into predictive computational models of the *M. tuberculosis* regulatory and metabolic networks [97].

1.7.4 Systems epidemiology

Systems epidemiology [98] aims to answer questions about the effect strain diversity on disease phenotypes. In tuberculosis, relatively little is known about strain-specific effects. Questions of this nature will be addressed by parallel high-throughput screens: individual strain sequencing combined with high-resolution host phenotyping using transcriptomics, proteomics, metabolomics and other approaches. One challenge of this approach will be to combine and integrate large-scale genomic data from the pathogen with phenotypic data from the host. One approach may utilise host transcriptomics to define the host response at high resolution, and to correlate the host states with pathogen genomics.

1.7.5 Interface of *M. tuberculosis* and host, and how this is modified

Tuberculosis is defined by the presence of *M. tuberculosis* in the host, and the subsequent immune response. In order for a response to occur, information about the pathogen has to reach the host via an afferent information collection system. This information is processed, and a response generated using effector components of the immune system. This flow of information from pathogen to host is crucial, and occurs via the interface between the host and pathogen. Defining the exact nature of this interface is required to understand host-pathogen information exchange. In addition to the siRNA approach outlined above [96] which identified host autophagy machinery as a crucial part of this interface, another method to understand this information exchange is the use of simultaneous host and pathogen transcriptomics, reviewed in [99]. This approach is challenging, as the pathogen RNA occurs in quantities orders of magnitude lower than host RNA. Array-based methods cannot routinely discriminate between different RNA isoforms, uncover unannotated non-coding

RNA sequences, define transcript borders, and suffer from a low signal-to-noise ratio due to non-specific and background hybridisation. In addition, RNA sample preparation is complex. RNAseq overcomes many of these problems. The method requires RNA to be extracted from samples containing both host cells and mycobacteria. This RNA is subjected to RNAseq, and resultant reads are mapped to both genomes, allowing inferences to be made about correlations of transcriptomic responses. As this technique relies on a stochastic sample of RNA from both host and organism, the absolute amount of RNA recovered from the organism is still much lower than that of the host; this has to be taken into account in the analysis of the dual RNAseq data. In order to partially alleviate this, one might employ techniques to specifically enrich for RNA of the under-represented organism; this is not usually recommended. Another strategy consists of depletion of ribosomal RNA (rRNA); this increases the relative information content of the prokaryotic RNA fraction. Again, this approach is not recommended if the sequencing depth can be increased sufficiently in order to identify transcripts of extremely low copy number. This, however, significantly increases costs.

1.7.6 Compartmentalisation and site of disease studies

Systems approaches may also be applied at the level of site of disease. For example, host transcriptomics may be performed in whole blood and site of disease simultaneously (Deffur et al, manuscript in preparation). For concurrent transcriptomics approaches to two compartments, care must be taken to account for profound differences in cell populations in the compartments (e.g. the site of disease is expected to be enriched for cell types germane to dealing with the infection whereas the whole blood compartment serves as the conduit for the efferent arm of the immune response [84]. One method of dealing with this is deconvolution of total compartment signatures into cell-specific signals [100] and performing cell-specific differential expression analyses [101].

1.7.7 Challenges

Some challenges faced by systems approaches are data integration and model integration. Data sets are generated on high-throughput platforms with unique readouts, and the difficulty becomes integrating the datasets at the level of the biological signal, and not the raw data, which contains both signal and noise. Additional integration challenges are file format conventions, different methods

of data storage (flat files, relation databases), the absence of metadata standards and the lack of precise, structured vocabularies that describe the system components exactly. Some existing models perform an adequate function in recreating emergent properties observed in either the host or the bacterium, making some simplifying assumptions about the other player in the two-component system. What is required is the integration of a complete model of mycobacterial metabolism and responses to stress and other perturbations within a multi-scale model of human immunology. In order for this to be accomplished, a complete description of the interface of host and pathogen, with associated transcriptional response networks is required.

1.7.8 Research priorities for systems approaches in co-infection with *Mycobacterium tuberculosis* and HIV

Define and delineate (new) phenotypes. Given the added complexity of phenotypes of co-infection with *M. tuberculosis* and HIV, the full range of phenotypes needs to be elucidated. This requires high throughput screens of healthy, immunosensitised and symptomatic individuals, combined with sensitive imaging techniques (e.g. PET/CT). High-resolution phenotyping is required in order to perform meaningful comparisons between groups of samples that measure thousands of parameters concurrently.

Controlled vocabularies. Terminology employed in clinical, animal model and molecular studies needs to be standardised, especially where higher order emergent phenomena are concerned. For example, there is lack of consensus in the literature for the appropriate use of terms like “dormant”, “latent” and “inactive” tuberculous lesions. On some occasions, different terms may mean the same thing, and on others, one term may refer to multiple phenomena. What is required is a strict one-to-one mapping of terms to biologic concepts. This is best accomplished by developing a controlled vocabulary (ontology), which can be used in annotating specific aspects of biological and computational models. Development of a TB ontology is underway under the auspices of a collaborative NHLBI-sponsored effort (TB Systems Biology).

What is the interface, and how does HIV change it? One of the main difficulties in integrating models of human immunology and mycobacterial biology is definition of the interface that allows bidirectional information flow between the two systems. In a way, this interface allows the

one system to "read" the internal state of the other system. Many features of this interface are known (ligands and receptors, signalling pathways), but others require elucidation (e.g. role of mycobacterial secreted factors).

Model integration. Controlled vocabularies and knowledge of the host-pathogen interface are both key to integrating computational models (e.g. metabolic models of *M. tuberculosis* and models of granuloma formation). Many current models consider one side of the host-pathogen relationship in considerable detail, while at the same time simplifying the biology of the other side. What is required is convincing interaction between host and pathogen that incorporates sufficient detail about both. Predictions of such models are again amenable to testing using either clinical samples or realistic animal models.

Create integrated models of HIV co-infection. Models of HIV-TB co-infection will require a significant increase in complexity, due to the increased phenotypic variability introduced by the variable immune dysfunction associated with HIV, and will ideally model all three organisms. They should exhibit known emergent properties of co-infection (e.g. effects of TB on HIV, effects of HIV on TB, immune reconstitution inflammatory syndrome).

1.8 Conclusion

Systems approaches to TB-HIV will complement on-going reductionist approaches by synthesis of models capable of predicting emergent phenomena related to this disease. In so doing, we will gain a deeper and more complete understanding of the drivers of pathogenetic processes, and will hopefully identify novel ways in which to interfere with this system in order to produce more favourable clinical outcomes. Systems approaches are challenging, and require a new ways of thinking about TB-HIV co-infection specifically, and infectious diseases research more generally[101], with the goal of ultimately redefining the host-pathogen research paradigm.

2 IMPI-MA: a two-compartment survey of the transcriptional landscape in HIV-TB

Chapter summary

In this chapter I describe IMPI-MA, the study that forms the basis for this thesis. First, the problem of tuberculous pericarditis in the context of a major HIV-TB epidemic is presented, as this disease presentation will serve as a two-compartment model of HIV-TB. Next, hypotheses and study questions are developed, followed by specific aims and objectives. Finally, I present the study design and outline my approach for dealing with a very heterogenous dataset.

2.1 Tuberculous pericarditis

2.1.1 Tuberculosis worldwide and in South Africa

Currently, approximately one third of the world's population is infected with *Mycobacterium tuberculosis*, and one in ten of those will develop clinically overt disease in their lifetime. In the most recent report on the global TB epidemic [102], the WHO provided the following statistics: In 2011, there were an estimated 8.7 million new cases of TB (13% co-infected with HIV) and 1.4 million people died from TB, including almost one million deaths among HIV-negative individuals and 430,000 among people who were HIV-positive. Worldwide incidence rates continue to fall very slowly. 5% of all cases of tuberculosis have multidrug resistant TB (MDR-TB).

In South Africa, the incidence of tuberculosis is amongst the highest in the world, at 993 cases per 100,000 population per year [102]. A high proportion of this disease burden is borne by HIV-infected individuals: approximately 70% of incident TB cases are HIV infected. Manifestations of

tuberculosis in HIV-infected individuals are highly variable, but an increased proportion of cases present with extrapulmonary tuberculosis.

2.1.2 Tuberculous pericarditis: phenotypes and complications

One of the less common but clinically very important presentations of tuberculosis is tuberculous pericarditis (TB-PC). Approximately 1% of cases have cardiac involvement. In the acute setting, tuberculous pericarditis presents with constitutional symptoms, chest pain and a pericardial effusion. Patients may present with cardiac tamponade requiring emergency drainage of the effusion.

Despite adequate antimycobacterial chemotherapy, up to 25% of patients with tuberculous pericarditis develop constrictive pericarditis, which has a high level of morbidity and 25% mortality at 6 months. The mortality rate rises to 40% at 6 months in patients with HIV infection who have clinical features of immunosuppression, and are not on antiretroviral therapy [103, 104]. This makes TB-PC the second-most lethal form of tuberculosis after tuberculous meningitis (TBM).

Strategies are needed to reduce the high complication rate in patients with TB-PC. The use of corticosteroids in tuberculous pericarditis is still controversial [105, 106], but is currently actively being studied in a randomised, controlled clinical trial.

TB-PC exhibits one of several distinct haemodynamic phenotypes. The phenotypes are identified using haemodynamic measurements obtained at the time of pericardiocentesis, and consist of the following:

1. Dry (acute) pericarditis
2. Purely effusive, with normalisation of right atrial pressure when the pericardium is drained to dryness
3. Effusive-constrictive, with evidence of visceral pericardial involvement and failure of right atrial pressure to normalise when the pericardium is drained to dryness [103]
4. Constrictive pericarditis (late-stage complication)

Understanding the sequence of events that leads to constrictive pericarditis might aid in developing tools to predict which patients with tuberculous pericarditis are at risk of developing constriction, and to find novel ways of preventing constriction by targeted disruption of constriction-inducing

pathways. The study, detailed below, in essence will try to elucidate biological processes that play a role in the pathogenesis of constrictive pericarditis.

It is clear that pericardial involvement in patients with tuberculosis presents a significant problem in South African healthcare setting. Better understanding of this disease is required in order to develop improved diagnostic and therapeutic modalities. In summary, TB-PC is second only to TB meningitis in terms of lethal potential, and it is an excellent model of extrapulmonary TB, with the ability to sample material directly from the disease site.

2.2 IMPI and related substudies

The Investigation of the Management of Pericarditis in Africa (IMPI Africa) registry [106] was established by the Cardiac Clinic, Department of Medicine at the University of Cape Town in order to study clinical, pathophysiological and outcome characteristics of a cohort of patients with tuberculous pericarditis.

Over the years since the original IMPI-Africa Registry was established, a number of studies have continued to collect data on patients with TB-PC. The broader registry can therefore be divided into several "phases" with different research focus.

1. Original IMPI registry
2. Registry 2
3. Clinical trial of adjunctive immunotherapy in TB-PC

In addition to the three sequential studies above, a number of substudies have sampled from the three populations above:

1. Haemodynamic phenotypes in TB-PC (Ntsekhe et al, manuscript in final review with PLOS ONE)
2. Immunology of tuberculous pericarditis
3. Diagnosis of tuberculous pericarditis
4. Microarray substudy (IMPI-MA) - the subject of this thesis

The complex relationship of study populations between studies and substudies is shown in Figure 2.1

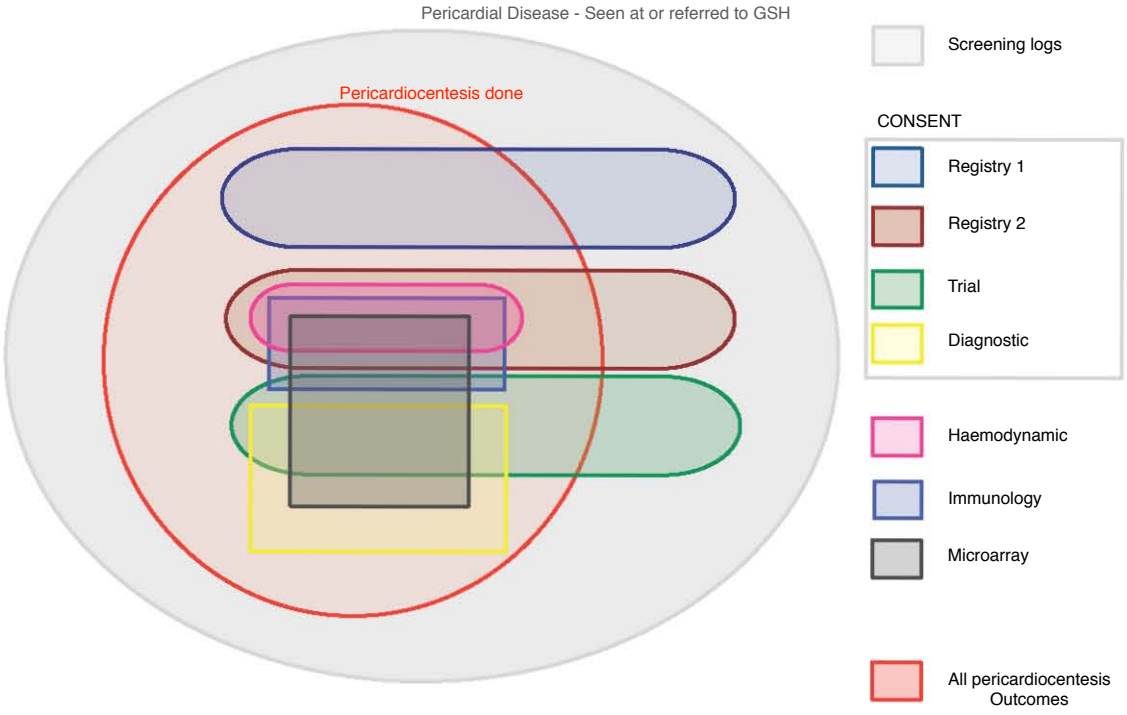


Figure 2.1: **Relationship between various IMPI cohorts.** The figure shows that three large cohorts (ovals) of patients were drawn from a the set of patients with TB-PC seen at Groote Schuur Hospital. A subset of patients presenting with TB-PC underwent pericardiocentesis (red circle). Four substudies were performed on overlapping patient populations (rectangles)

2.2.1 Immunology substudy

A collaboration between the Cardiology group leading the IMPI projects and the laboratory of Professor Robert J Wilkinson at the IIDMM, University of Cape Town, has been active since 2006. In work for her Masters and PhD degrees, Kerryn Matthews, together with her supervisor Katalin A Wilkinson, has performed extensive immunological characterisation of samples obtained from individuals with TB-PC

The following assays were performed:

1. Restimulation assay of whole blood and pericardial fluid using ESAT-6 and CFP-10 with measurement of IFN-γ in the supernatant

2. ELISPOT analysis of peripheral blood mononuclear cells (PBMC) and pericardial fluid cells (PFC) for IFN-γ using multiple mycobacterial antigens
3. Surface phenotyping of T-cells from PBMC and PFC using fluorescence-activated cell sorting (FACS)
4. Quantitative RT-PCR (TaqMan) for forty-two genes involved in inflammation and regulation of inflammation
5. Measurement of twenty-four cytokines and other mediators involved in inflammation, fibrosis and tissue destruction in serum and cell-free pericardial fluid using ELISA and luminex

Key findings reported in her PhD thesis [107]

1. Increased recognition of *Mycobacterium tuberculosis*-specific antigens by T cells in the pericardial fluid with a phenomenon of "Pericardial" and "Peripheral dominance" in the recognition of *M. tuberculosis*-specific antigens
2. There are more differentiated antigen experienced CD4+ and CD8+ T cells in HIV-1 uninfected TB-PC patients, while HIV skews the phenotype of memory T cells in the pericardium towards less differentiated CD4+ and CD8+ T cells
3. HIV skews the cell-specific cytokine response towards a more polyfunctional T cell profile, probably contributing to pathology of TB pericarditis, rather than protection
4. The pericardial space has been shown to be largely pro-inflammatory, but it is also profibrotic, with compartmentalisation of pro-collagen genes and lack of fibrotic regulators, irrespective of the HIV status of the patients (see Chapter 5 for additional analysis of this data)
5. There is dysregulation between MMP and TIMP molecule production and binding, which may overstimulate the synthesis of inflammatory and profibrotic cytokines

2.2.2 Haemodynamic phenotypes

In a study submitted to PLOS ONE, Mpiko Ntsekhe, et al report a high prevalence (53%) of the effusive-constrictive phenotype of TB-PC in the study population, and this phenotype is accom-

panied by high interleukin 10 (IL-10) levels in blood and pericardial fluid. This suggests that the inflammatory and anti-inflammatory milieu may be differentially regulated in effusive and effusive-constrictive TB-PC.

2.2.3 Synthesis

The findings from the above work suggest the following:

1. Cellular profiles in blood and pericardial fluid in TB-PC may differ significantly between HIV-1-infected and -uninfected individuals. This may be detectable as differential gene expression at whole sample and cell-specific level

2. Several components of the immune response to tuberculosis may be compartmentalised as evidenced by differential mRNA transcript, protein levels and cell population proportions between blood and pericardial fluid

3. In tuberculous pericarditis, the immune response may be differentially regulated based on the haemodynamic phenotype of the pericardial effusion (in blood, pericardial fluid, or both)

2.3 Development of a systems approach

The finding of an interferon-inducible neutrophil-driven blood transcriptional signature in human tuberculosis [82] in HIV-1 uninfected study subjects argues strongly for replicating this approach in an HIV-1 infected cohort. In addition, site of disease data as produced by Matthews et al and Ntsekhe et al argue strongly that gene-expression patterns underlying cellular and cytokine responses at the site of disease are amenable to discovery and may shed significant light on site-of-disease pathogenetic mechanisms in HIV-TB. I therefore embarked on developing a theoretical framework within which questions relating gene expression in tuberculosis, HIV and body compartment could be answered, and the results compared.

In this context, it is worth briefly discussing what tissue-level transcriptomes actually represent. The transcriptome is the set of all RNA molecules, including mRNA, rRNA, tRNA, and other non-coding RNA produced in one or a population of cells (http://en.wikipedia.org/wiki/

Transcriptome). It can be measured using array-based and sequencing-based techniques. Importantly, the transcriptome varies dynamically over time, and is strongly influenced by environmental factors. The unit of measure for the transcriptome is "transcript abundance", which is influenced by factors such as constitutive expression, regulation of the activity of individual genes due to signal processing within cells and stochastic processes. Individual types of cells exhibit a predilection to express particular genes which, taken together, define and establish the specific properties of the cell type.

2.4 Hypotheses and study questions

Three main hypotheses underlie this thesis.

1. Active tuberculosis is distinguishable from not active tuberculosis on the basis of differential gene expression in both HIV-1 uninfected and HIV-1 infected individuals
2. HIV-1 infection is distinguishable from absence of HIV-1 infection on the basis of differential gene expression in multiple health and disease contexts, including active tuberculosis
3. Differential gene expression between matched samples of blood and pericardial fluid is due to both altered cell proportions between the compartments as well as cell-specific differential gene expression

2.5 Aims and objectives

2.5.1 Aims

The broad aim of this work is to provide a survey of the transcriptional landscape of HIV-TB in a two-compartment model at the level of individual genes, gene modules and individual cell types.

2.5.2 Objectives

1. Retrospective selection and prospective collection of matched blood and pericardial fluid samples from individuals with suspected or confirmed TB-PC, and of blood samples of three classes of controls (healthy, LTBI and pulmonary tuberculosis (PTB))

2. Retrospective curation, prospective collection and synthesis of clinical and phenotype data relating to the above samples

3. Measurement of host transcript abundance in pericardial fluid (PCF) and blood using cDNA microarrays

4. Development of an analytic framework in the R programming language that allows for addressing the stated hypotheses and additional research questions. Importantly, results must be output at gene, gene module and cellular levels

5. Placement of differentially expressed genes and modules in biological context by examining gene modules in co-expression networks as well as biological pathways

2.6 Overall study design

This study, though complex in analytic design, is an observational study of a heterogenous cohort of individuals, each of whom had contributed either a blood or matched blood and PCF samples, depending on disease status. Throughout this manuscript, this study will be referred to as IMPI-MA (**IMPI**-Africa registry **M**icro**A**rray substudy).

2.6.1 Contrasts and comparisons

Two concepts are important to differentiate at this point:

1. Contrast: a contrast is defined as a phenotypic variable that defines exactly two states between which differential gene expression is assessed. The full set of contrasts will be described and justified in Chapter 8, where the full dataset and the associated phenotypes will be described. The three contrasts defined *a priori* (as implied by the hypotheses) are:

    a) **TB-status**, consisting of *active tuberculosis* (either PTB or TB-PC) or *not active tuberculosis* (either healthy individuals or asymptomatic individuals with evidence of prior sensitisation to *Mycobacterium tuberculosis*)

    b) **HIV-status**, consisting of *HIV-1 infected* and *HIV-1 uninfected* individuals, respectively

c) **Compartment**, consisting of whole *venous blood* or whole *pericardial fluid* samples

2. Comparison: a comparison is defined as an assessment of similarity or difference in two or more groups following assessment of one particular contrast, based on another phenotypic variable, termed the *context variable*. For example, differential expression analysis may be performed for the contrast TB-status in two sample subsets (HIV-1 infected and HIV-1 uninfected), and the results compared. Conversely, HIV-status may be used as the contrast, and *active tuberculosis* and *not active tuberculosis* status used as basis for the comparison. Figure 2.2 shows this graphically.

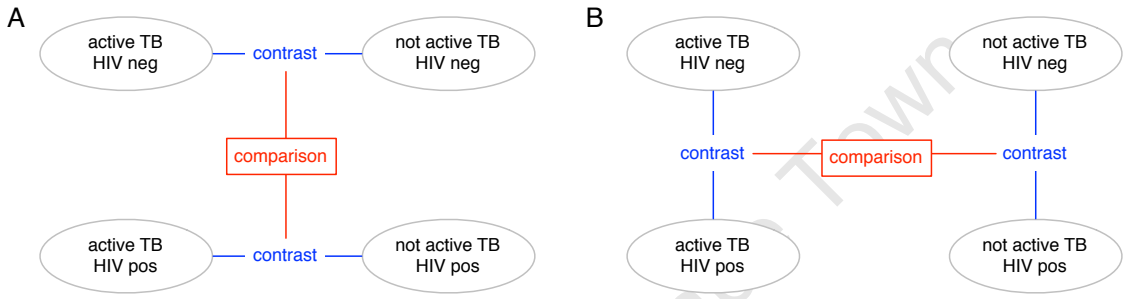


Figure 2.2: **The difference between the concepts *Contrast* and *Comparison*.** In panel A, the contrast *TB-status* is used to discover a list of differentially expressed genes in two populations. The two gene lists can then be compared on the basis of HIV-status. In panel B, the converse experiment is shown, with *HIV-status* being the contrast and TB-status the comparator.

2.6.2 Phenotype space as hypercube

Given the three contrasts defined above, we can visualise each sample represented mapped to a corner of a three-dimensional cube. Examination of the edges of the cube, shown in Figure 2.3 reveals all possible contrasts and comparisons involving these three parameters. This concept will be expanded upon in Chapter 8, where we will generalise the parameter space for the full analysis to a subgraph of a six-dimensional hypercube. The cubic graphical parameter space is analogous to a three-dimensional space showing principal components of a dataset defined by three parameters. In an ideal situation, the point clouds for each of the eight groups would be arranged in a cube-like fashion, with the eight centroids forming the vertices of a cube.

The three-dimensional cube immediately suggests a design for an analytic algorithm that will

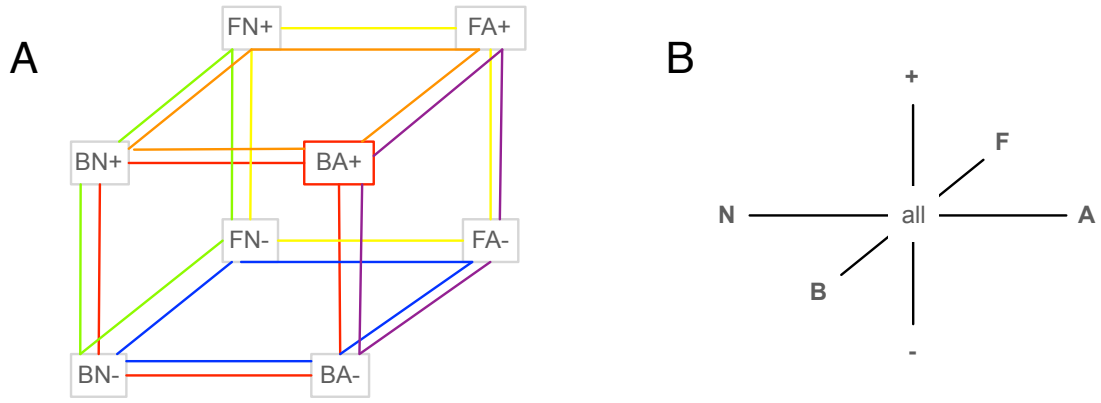


Figure 2.3: **3D parameter space.** Panel A. Three-dimensional cube representing the main parameter space required to address the three stated hypotheses is shown. Abbreviations: **B** blood; **F** pericardial fluid; **A** active tuberculosis; **N** not active tuberculosis; **+** HIV-1 infected; **-** HIV-1 uninfected. There are 2^3=8 combinations of the three variables, yielding the eight corners of the cube as shown. Given the eight corners, there are a total of 12 potential contrasts and 6 potential comparisons (See Figure 2.2). Each face of the cube corresponds to one of 2×3=6 views of the dataset. Panel B shows the three orthogonal contrasts used to address the main hypotheses of this thesis.

potentially demonstrate a complete survey of the transcriptional landscape of HIV-TB in a two-compartment model.

3 Materials and methods

Chapter summary

In this chapter I describe generation and analysis of preliminary data based on an RT-PCR approach, IMPI-MA study subjects, the clinical phenotype database, RNA sample processing, further processing of RNA samples for microarray, microarray data generation, microarray data management and a general approach to microarray data analysis as developed in R.

3.1 Preliminary data: RT-PCR data generation and analysis

One subset of the TB-PC immunology data will be discussed here, as I will present an analysis of this data in Chapter 4. In work done prior to the project that is the subject of this thesis, Kerryn Matthews studied matched blood and pericardial fluid samples from individuals with TB-PC by quantitative real-time polymerase chain reaction (qRT-PCR). The generation of this data was performed in 2006-2010, and the analysis reported in her PhD thesis [107]. For the purposes of developing an analytic framework for the microarray study, I re-analysed this dataset using a workflow designed for micorarray data. Here I briefly review the laboratory work done by Kerryn Matthews, and describe the analytic pipeline.

3.1.1 RNA collection and processing

Three mL of matched blood and pericardial fluid samples were collected in PAXgene Blood RNA tubes (PreAnalytiX GmbH, CH, catalog number 762165) at the time of pericardiocentesis, and frozen at $-20\,^{\circ}\mathrm{C}$ pending RNA extraction.

3.1.2 Data generation[1]

All of the following work was performed by Kerryn Matthews and is hereby fully acknowledged. Forty-two genes were chosen to evaluate the host transcriptional response in two compartments based on known biological significance in tuberculosis or HIV pathogenesis. These genes are listed in Table 3.1. RNA was extracted using the PAXgene Blood RNA Kit (PreAnalytiX, catalog number 762174) according to the manufacturer's instructions, and stored at $-80\,^{\circ}$C. Quantity and quality of the total RNA was determined using a NanoDrop (ND1000, Thermo Scientific, USA). RNA was reverse transcribed to cDNA using the Quantitect Reverse Transcription kit (Qiagen, USA, catalog number 205313) and a total of 1μg/ml of cDNA was used in each RT-PCR reaction. Primers and probes for RT-PCR were purchased from Applied Biosystems as predesigned inventoried assay reagents. The TaqMan Gene Expression Assays (Applied Biosystems, USA) used to determine the gene expression of forty-two genes are listed in Table 3.1, and the experiments were performed in a 25μl final reaction volume under universal cycling conditions on the ABI 7000 platform (Applied Biosystems, USA) as described [30]. Human beta-actin was used as the single internal endogenous control for each sample assayed. All data values output are ΔCT values, calculated by subtracting the cycle threshold of beta-actin from the cycle threshold of the gene of interest. See Figure 3.1 for illustration of this method. On occasions where the gene of interest failed to amplify after forty cycles, the ΔCT value was recorded as "invalid" by the instrument.

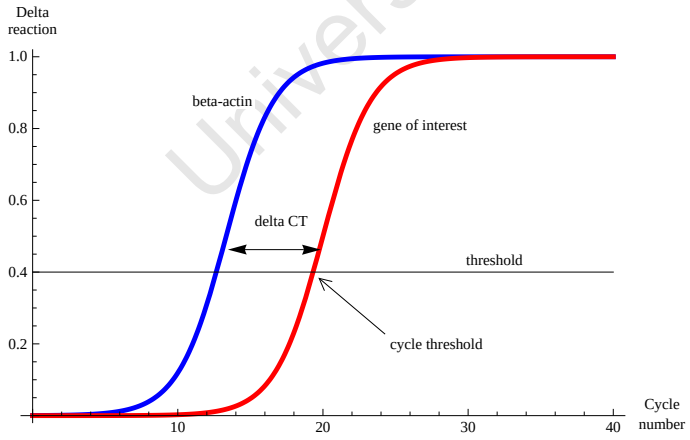


Figure 3.1: **RT-PCR: calculation of** ΔCT

[1]This subsection is based on original input by Kerryn Matthews

Table 3.1: **42 genes selected for RT-PCR analysis**

HGNC Symbol	HGNC Approved Name	HGNC ID	Entrez Gene ID	Protein name
IL1B	interleukin 1, beta	5992	3553	Interleukin-1 beta
IL2	interleukin 2	6001	3558	Interleukin-2
IL6	interleukin 6	6018	3569	Interleukin-6
IL18	interleukin 18	5986	3606	Interleukin-18
IL24	interleukin 24	11346	11009	Interleukin-24
TNF	tumor necrosis factor	11892	7124	Tumor necrosis factor
IFNG	interferon, gamma	5438	3458	Interferon gamma
CXCL10	chemokine (C-X-C motif) ligand 10	10637	3627	C-X-C motif chemokine 10
IL4	interleukin 4	6014	3565	Interleukin-4
IL13	interleukin 13	5973	3596	Interleukin-13
IL17A	interleukin 17A	5981	3605	Interleukin-17A
IL22	interleukin 22	14900	50616	Interleukin-22
IL23A	interleukin 23, alpha subunit p19	15488	51561	Interleukin-23 subunit alpha
TGFB1	transforming growth factor, beta 1	11766	7040	Transforming growth factor beta-1
IL10	interleukin 10	5962	3586	Interleukin-10
FOXP3	forkhead box P3	6106	50943	Forkhead box protein 3
IL8	Interleukin 8	6025	3576	Interleukin-8
LCN2	lipocalin 2	6526	3934	Neutrophil gelatinase-associated lipocalin
CAMP	cathelicidin antimicrobial peptide	1472	820	Cathelicidin antimicrobial peptide
DEFB4A	defensin, beta 4A	2767	1673	Beta-defensin 4A
ELANE	elastase, neutrophil expressed	3309	1991	Neutrophil elastase
EGF	epidermal growth factor	3229	1950	Pro-epidermal growth factor
CTGF	connective tissue growth factor	2500	1490	Connective tissue growth factor
CSF2	colony stimulating factor 2 (granulocyte-macrophage)	2434	1437	Granulocyte-macrophage colony-stimulating factor
IL9	interleukin 9	6029	3578	Interleukin-9
COL1A1	collagen, type I, alpha 1	2197	1277	Collagen alpha-1(I) chain
COL1A2	collagen, type I, alpha 2	2198	1278	Collagen alpha-2(I) chain
COL4A1	collagen, type IV, alpha 1	2202	1282	Collagen alpha-1(IV) chain
COL4A2	collagen, type IV, alpha 2	2203	2203	Collagen alpha-2(IV) chain
ARG1	arginase, liver	663	383	Arginase-1
SPARC	secreted protein, acidic, cysteine-rich (osteonectin)	11219	6678	SPARC
MMP1	matrix metallopeptidase 1 (interstitial collagenase)	7155	4312	Interstitial collagenase
MMP2	matrix metallopeptidase 2 (gelatinase A, 72kDa gelatinase, 72kDa type IV collagenase)	7166	4313	72 kDa type IV collagenase
MMP3	matrix metallopeptidase 3 (stromelysin 1, progelatinase)	7173	4314	Stromelysin-1
MMP7	matrix metallopeptidase 7 (matrilysin, uterine)	7174	4316	Matrilysin
MMP8	matrix metallopeptidase 8 (neutrophil collagenase)	7175	4317	Neutrophil collagenase
MMP9	matrix metallopeptidase 9 (gelatinase B, 92kDa gelatinase, 92kDa type IV collagenase)	7176	4318	Matrix metalloproteinase-9
MMP10	matrix metallopeptidase 10 (stromelysin 2)	7156	4319	Stromelysin-2
MMP11	matrix metallopeptidase 11 (stromelysin 3)	7157	4320	Stromelysin-3
TIMP1	TIMP metallopeptidase inhibitor 1	11820	7076	Metalloproteinase inhibitor 1
TIMP2	TIMP metallopeptidase inhibitor 2	11821	7077	Metalloproteinase inhibitor 2
TIMP3	TIMP metallopeptidase inhibitor 3	11822	7078	Metalloproteinase inhibitor 3

3.1.3 Analysis pipeline

In order to support development of an analysis pipeline for the microarray data, I developed a simplified version of this pipeline to deal with the RT-PCR data as this data was analogous to microarray data, but far less complex (by almost four orders of magnitude).

Raw RT-PCR data was expressed as delta cycle threshold (ΔCT) in both blood and pericardial fluid samples. In order to emulate data generated in microarray experiments, the ΔCT values were transformed by $\log_{10} \times 2^{\Delta CT}$, which yielded expression levels *relative* to beta-actin on a continuous log scale, with smaller values indicating lower expression levels. No further normalisation was performed, as each value was already normalised to beta-actin, and assumptions that apply to large data sets of thousands of genes, as found in typical microarray experiments can not be assumed to hold true for forty-two individually selected genes. Several genes failed to reach the detection threshold by 40 cycles, and a ΔCT value of 40 was assigned in those cases, representing extremely low levels of gene expression.

All downstream data analysis was performed in R [108], using packages from the bioconductor suite (http://www.bioconductor.org), and additional packages from CRAN (http://cran.r-project.org). The matrix of expression values was clustered by hierarchical clustering with Euclidean distance and complete linkage. Correlation of gene expression in both blood and pericardial fluid was assessed by computing the matrix of pairwise Pearson correlation coefficients. Genes that had very low expression levels (ΔCT value > 38 in more than 5% of samples) were removed from the analysis by a non-specific filtering step.

Differential gene expression between blood and pericardial fluid was determined using a linear model approach implemented in R using the package *limma* [109]. In short, log-ratios were estimated between two or more target RNA samples simultaneously by performing least-squares regression modelling for each gene. The standard errors of each model are moderated using an empirical Bayes model. The output of *limma* was a moderated t-statistic and a log-odds of differential expression for each gene. The resulting log-fold changes (differential expression) and the corresponding statistical significance were visualised using a volcano plot. Visualisation of various gene subsets by principal component analysis was performed using the ClassDiscovery package from [110]. Heatmaps were calibrated by including one column each of minimum and maximum values

of the raw data. This preserves the colour hue and intensity of each datapoint across all heatmaps. All values were clustered using hierarchical clustering with average linkage.

3.2 IMPI-MA study subjects

In order to address the three hypotheses above, I required RNA samples from "active tuberculosis" cases and "not active tuberculosis controls", stratified by HIV status and compartment (blood and pericardial fluid). Instead of attempting to prospectively recruit adequate numbers of study subjects, I had the opportunity to utilise samples collected prior to the start of the IMPI-MA project, referred to below as "retrospective cases" and "retrospective controls". The study subjects and their respective phenotype data and RNA samples are described in detail. Even with a large retrospective sample collection, it became apparent that prospective study subjects and samples were required to reach adequate numbers of samples. Therefore, an additional cohort of prospectively collected samples was generated.

3.2.1 Definitions of cases and controls

Specific definitions apply for certain terms used in this thesis, including case definitions of disease states. These are summarised in Table 3.2.

Table 3.2: **Definitions of terms.** Abbreviations: ADA: Adenosine deaminase, CXR: Chest X-Ray. *Tygerberg score* refers to a numeric score based on clinical (fever, weight loss and night sweats) and laboratory (serum globulin and leukocyte count) [111]

Term	Criterion	Value
TB Status	Not active tuberculosis or Active tuberculosis	
TB Status: Not active tuberculosis	Healthy or Latent TB infection	
Healthy	Symptoms, TST and IGRA	Asymptomatic, and no evidence of prior immune sensitisation to *Mycobacterium tuberculosis* (TST negative and IGRA negative)
Latent TB infection	Symptoms, TST and IGRA	Asymptomatic, and evidence of prior immune sensitisation to *Mycobacterium tuberculosis* (TST positive or IGRA positive)
TB Status: Active tuberculosis	PTB (definite) or TBPC (definite)	
PTB (definite)	Demonstration of Mycobacterium tuberculosis in sputum	Sputum smear positive for acid-fast bacilli
		OR Positive culture or PCR for *Mycobacterium tuberculosis* in pericardial fluid
PTB (probable)	Symptoms, CXR	Symptoms and signs of PTB; compatible CXR changes; response to therapy
TB-PC (definite)	Demonstration of Mycobacterium tuberculosis in pericardial fluid	Positive culture or PCR for *Mycobacterium tuberculosis* in pericardial fluid
TB-PC (probable)	Pericardial effusion AND	Demonstration of significant pericardial effusion on echocardiogram
	No evidence of pericardial infection with other bacteria AND	Negative bacterial culture of PCF
	Elevated adenosine deaminase level in pericardial fluid OR	ADA > 40 U/L in PCF
	Elevated IFNγ level in pericardial fluid OR	IFNγ > 50 pg/ml in PCF
	Good clinical response to anti-TB chemotherapy OR	Clinical assessment of response to therapy
	Demonstration of Mycobacterium tuberculosis in a site other than PCF OR	Positive smear, culture or PCR for Mycobacterium tuberculosis at a site other than PCF
	Tygerberg score	A Tygerberg score of 6 or greater
Effusive-constrictive TBPC	**Pre-pericardiocentesis right atrial pressure** elevated AND	> 8 mmHg
	low **transmural pressure gradient** AND	$\leq$ 4 mmHg
	Failure of **Post-pericardiocentesis RAP** to fall to normal levels	>11 mmHg
Effusive TBPC	**Post-pericardiocentesis RAP** falls to normal levels and the post-pericardiocentesis intrapericardial pressure falls to to zero	
HIV Status	HIV-1 infected or HIV-1 uninfected	
HIV-1 infected	ELISA for HIV-1	ELISA for HIV-1 reactive
HIV-1 uninfected	ELISA for HIV-1	ELISA for HIV-1 non-reactive
Compartment	Blood or Pericardial Fluid	
Blood		Venous blood
Pericardial fluid		Pericardial fluid collected at pericardiocentesis

3.2.2 Retrospective TB-PC cases

Starting in 2006, Kerryn Matthews in the laboratory of Robert J Wilkinson at CIDRI, IIDMM, UCT started collecting matched blood and pericardial fluid samples form patients being enrolled in Registry 2 of the IMPI-Africa Registry by Mpiko Ntsekhe and colleagues. This specific sample collection period ended in 2010 when the Biosafety Level 3 (BSL-3) facility at the IIDMM was upgraded and all work requiring BSL-3 laboratory conditions ceased.

The majority of subjects during this time were also enrolled in a "Haemodynamic Phenotype" substudy of the IMPI-Africa Registry. The clinical workflow is described in Figure 3.2. Patients were included in the Registry if they were 18 years or older, gave written informed consent, had an effusion of at least 500mm^3 (at least 10 mm free fluid at the cardiac apex on echocardiogram) and had tuberculosis suspected or confirmed as aetiology for their effusion. In addition patients were included in the haemodynamic phenotype substudy if right atrial pressures pre- and post pericardiocentesis were available.

1. Screening: Echocardiogram confirms pericardial effusion
2. Patient consented and enrolled in the IMPI-Africa Registry
3. Pericardiocentesis under fluoroscopic guidance in the Heart Catheterisation Laboratory at Groote Schuur Hospital
    a) Effusion tapped to dryness
    b) Blood and pericardial fluid samples collected
    c) In a subset of cases, simultaneous right-heart catheterisation was performed in order to obtain the measurements required for categorisation of haemodynamic phenotype
4. Phenotyping
    a) History and clinical examination
    b) Appropriate microbiological tests to detect *Mycobacterium tuberculosis*
    c) Additional biochemical and other tests on blood

Figure 3.2: **Clinical workflow for retrospective TB-PC cases**

3.2.3 Retrospective controls

The ILULU partnership, an EU-funded consortium aiming to develop a novel diagnostic marker for tuberculosis, enrolled children and adults in study sites in South Africa and Malawi in six clinical phenotypes:

1. TB+ HIV-
2. TB+ HIV+
3. LTBI (Latent TB) + HIV+
4. LTBI+ HIV-
5. HIV+ with opportunistic infections other than TB
6. HIV- with acute and chronic infections other than TB

Following completion of the first phase of this study, hereafter referred to as the "EU study" (during which cDNA microarrays were used to generate transcriptomic data), duplicate RNA samples remaining at the study sites were available for subsequent use at the discretion of the site principal investigator. I obtained permission from Professor Robert J Wilkinson, principal investigator for the adult study in Cape Town, to utilise selected samples based on needs for positive and negative controls for the IMPI-MA study.

The study subjects for the adult Cape Town arm of the study were enrolled at Ubuntu Clinic, Site B, Khayelitsha, Cape Town. The workflow of enrolment is described in Figure 3.3. Multiple asymptomatic subjects were screened for inclusion into the LTBI arms of the study; a number of these subjects turned out to have negative TST and IGRA responses, and while they were not included in the ILULU consortium study, their samples remained available for study. These constituted the "healthy" controls (see Table 3.2). LTBI+ HIV+ and LTBI+HIV- constituted the "LTBI" controls. Lastly, TB+ HIV+ and TB+ HIV- constituted the "PTB" controls.

3.2.4 Prospective TB-PC cases

Following completion of refurbishment work at the BSL-3 facility at the IIDMM, prospective sample collection could be restarted. From March 2011 to February 2012 I collected blood and pericardial fluid samples from patients enrolled in the diagnostic substudy of the ongoing IMPI

1. Screening: History and clinical examination
2. HIV counselling and testing
    a) If HIV-1 infected: CD4 count
    b) If HIV-1 uninfected: no CD4 count
3. TST and IGRA performed on asymptomatic patients
4. Investigation of symptomatic patients
    a) CXR
    b) Sputum for AFB and TB culture
    c) Other tests as indicated (specifically to detect other infections or malignancies)
5. Phenotyping based on results of 1-4

Figure 3.3: Clinical workflow for retrospective controls

Registry and/or the IMPI trial, a prospective clinical trial of immunotherapy in tuberculous pericarditis.

1. Screening: Echocardiogram confirms pericardial effusion
2. Patient consented and enrolled in the IMPI-Africa Registry: Diagnostic substudy and/ or IMPI-trial
3. Pericardiocentesis under fluoroscopic guidance in the Heart Catheterisation Laboratory at Groote Schuur Hospital
    a) Effusion tapped to dryness
    b) Blood and pericardial fluid samples collected
4. Phenotyping
    a) History and clinical examination
    b) Appropriate microbiological tests to detect *Mycobacterium tuberculosis*
    c) Additional biochemical and other tests on blood

Figure 3.4: **Clinical workflow for prospective TB-PC cases**

3.2.5 Ethics clearance for the various cohorts

All subjects provided written informed consent. This included sample storage for subsequent analysis. All subjects with TB-PC who were recruited into one of the IMPI cohorts are covered by UCT HREC REF: 102/2003, 402/2005 & 289/2007, while all subjects recruited by the EU study are covered by UCT HREC REF: 012/2007.

3.3 RNA and other Samples

3.3.1 Retrospective TB-PC cases

For each subject, 3 mL blood and pericardial fluid were collected in PAXgene containers. Following manufacturer's guidelines, the PAXgene tubes were frozen at $-20\,^{\circ}$C. Subsequently, RNA was extracted using the PAXgene Blood RNA Kit (PreAnalytiX, catalog number 762174) according to the manufacturer's instructions, and stored at $-80\,^{\circ}$C. RNA of 28 matched blood and pericardial fluid samples was used in the RT-PCR assay described in Section 3.1, and the remainder of extracted RNA was kept frozen at $-80\,^{\circ}$C. RNA yield and quality was assessed using NanoDrop 1000 and RIN as determined by AB Bioanalyzer 2100 (Agilent Technologies, USA).

Other blood and pericardial fluid samples were collected, processed and analysed as previously described [107, 112, 49].

3.3.2 Retrospective controls

For each subject, 3 mL blood was collected in PAXgene containers, respectively. Following manufacturer's guidelines, the PAXgene tubes were frozen at $-20\,^{\circ}$C. Subsequently, RNA was extracted using PAXgene Blood RNA Kit and the extracted RNA was kept frozen at $-80\,^{\circ}$C. RNA yield and quality was assessed using NanoDrop.

3.3.3 Prospective TB-PC cases

All work below was performed by myself in the BSL-3 facility at the IIDMM.

RNA samples For each subject, 3 mL blood and pericardial fluid were collected in PAXgene containers, respectively. Following manufacturer's guidelines, the PAXgene tubes were frozen at $-20\,^{\circ}$C after being allowed to stand undisturbed for at least two hours. Subsequently, RNA was extracted using PAXgene Blood RNA Kit. The remainder of extracted RNA was kept frozen at $-80\,^{\circ}$C. RNA yield and quality was assessed using NanoDrop (NanoDrop 2000c, Thermo Scientific, USA).

Serum For each subject, venous blood was collected in 2×5.0mL SST (BD, USA, Catalog number 367955) tubes. Serum was separated using centrifugation of the SST tube at 3000 rpm for 10 minutes with the brake on; serum the above gel layer was removed by pipette and and transferred to a Nunc CryoTube Vial (Thermo Scientific, catalog number 375418) (max. volume per vial = 1.8 mL) using a filter tip. Labeled cryotubes were frozen at $-80\,^{\circ}$C and recorded in the freezer log.

Separation of PBMC For each subject, venous blood was collected in 2×9.0mL VACUETTE Plasma Sodium Heparin tubes (greiner bio-one, USA, catalog number 455051). Peripheral blood mononuclear cells (PBMC) were separated from plasma using the Ficoll method. Blood was mixed with an equal volume of Dulbecco's Phosphate-buffered saline (PBS) (Sigma-Aldrich Life Science, USA, catalog number D8537) and carefully layered onto Ficoll-Paque (GE Healthcare, USA, catalog number 17-1440-03) in a 50 mL conical tube (BD Falcon, USA, catalog number 352070). The blood/PBS/Ficoll-Paque layered mixture was then centrifuged at 700g for 20 minutes with the brakes off. Following this, the buffy coat containing PBMC (see Figure 3.5) was removed by Pasteur pipette and transferred into 15mL RPMI-1640 (Sigma-Aldrich Life Science, USA, catalog number R8758) medium, which was then made up to 40 mL for the first wash step by centrifugation at 600 x g for 10 minutes (brakes on). After this, the supernatant was discarded, and the pellet resuspended with 10 mL RPMI containing 10% heat-inactivated foetal calf serum made up from FBS Superior stock solution (RPMI+10%FCS) (Biochrom, Germany, catalog number S 0615). The cells were counted twice using standard counting chambers using Trypan Blue (Sigma-Aldrich Life Science, USA, catalog number T8154) and Crystal Violet (Sigma-Aldrich Life Science, USA, catalog number C3886-25G), respectively, and the results averaged. The former allows to distinguish live from dead cells, and the latter lyses erythrocytes which may still be present). The cells were

finally centrifuged at 600 x g for 10 minutes with brakes on, and frozen as described below.

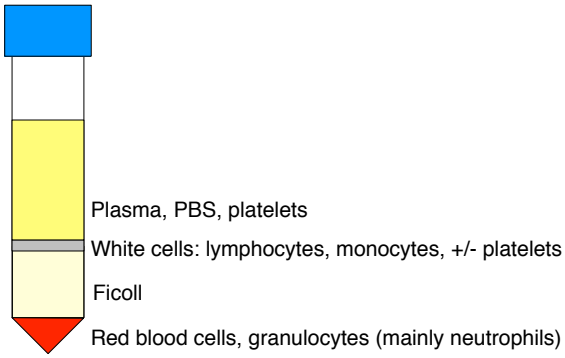


Figure 3.5: **Layering of components after Ficoll separation**

PFC and Cell-free pericardial fluid The method used for separating pericardial fluid cells (PFC) from pericardial fluid depended on blood content of the pericardial fluid. In some cases there was gross blood admixture, and the fluid resembled blood macroscopically, and in other cases the fluid was clear, serous fluid.

In case of no or minimal visible blood contamination, the contents of all three sodium heparin tubes containing pericardial fluid were transferred to a sterile 50 mL conical tube and centrifuged at 600 x g for 10 minutes with the brakes off. The supernatant, cell-free fluid (CFF), was removed by pipette and transferred to a cryovial (max. volume per vial = 1.8 mL); in most cases I aimed to freeze 3 x 1.8 mL. The remainder of the supernatant was discarded. The pellet containing PFCs was resuspended in 40 mL RPMI and centrifuged in the first wash step at 600 x g for 10 minutes with the brakes on. Supernatant was discarded and 10 mL of RPMI+10% HI-FCS added to the cells, which were then counted as described above. The final wash step consisted of centrifuging at 600 x g for 10 minutes with the brakes on, and the cells were frozen as described below.

In case of gross blood contamination (i.e. blood visible to the naked eye), the Ficoll method was used as above in order to cope with the large number of red cells, with the following modification: two of the three tubes were transferred to a 50 mL conical tube *without* addition of PBS, and the remainder of the steps were identical to the ones above in the Ficoll procedure for separating PBMC from blood. The remaining sample tube was processed by the standard PCF method in order to recover cell free fluid; the cells obtained in this process were discarded.

Freezing procedure for PBMC and PFCs After the second wash step in RPMI+10%HI-FCS, the supernatant was discarded. $1-10\times10^6$ cells (PBMC or PFC) were resuspended in 1 mL of freezing solution by carefully adding 0.4 mL of RPMI+10% HI-FCS to the pellet with a pipette, mixing carefully, and adding 0.5 mL of freezing medium (20% dimethylsulfoxide in 10% HI-FCS) drop-wise, while gently agitating the mixture. The cell/ cell-freeze mix was transferred to a labelled cryovial which was in turn transferred to a Mr Frosty and frozen at $-80\,^{\circ}$C overnight. After overnight storage, the cells were transferred to the vapour phase of liquid nitrogen for long-term storage, and the storage log completed.

3.3.4 Summary: RNA samples and banked specimens

RNA samples from blood, and in cases with TB-PC also pericardial fluid, were collected in a uniform way as per manufacturer guidelines, stored in identical fashion prior to extraction and extracted using identical kits. This addresses one potential source of technical bias in microarray studies: use of different RNA collection, storage and extraction methods. Banked serum, cell free fluid, PBMC and PFCs are available for future validation work.

3.4 Clinical phenotype database

Clinical phenotype data for the the RNA samples used in the IMPI-MA study was obtained by multiple investigators over a five year period. All retrospective data was obtained in electronic format from the investigators who had collected it, and I captured and collated all prospectively collected clinical data (SET-2).

3.4.1 Data sets

Data sets are listed in Table 3.3. The data sets are heterogenous, as each was originally developed for a specific purpose other than provision of clinical phenotype data for IMPI-MA. These datasets required many hours of work to produce, and have been used in supporting numerous publications, and will continue to support others. For the purposes of IMPI-MA, I required one dataset which used identical variables for parameters of interest. This was to be achieved by integrating the data, ensuring the ability to consistently phenotype all samples used in the microarray analysis.

Table 3.3: **Sources of clinical phenotype data used in IMPI-MA**

Name	ID	Investigator	Time period	N	Description of data types	Format	Challenges
IMPI phase 2 comprehensive	1	Mipko Ntsekhe, Faisal Syed	22/1/2006 - 13/8/2008	85	ID,demographic, clinical, clinical laboratory, echocardiography, ECG, microbiology, cytokine, haemodynamic and outcome data	xls	Mac/ Windows date error
IMPI phase 2 ECP plus lab variables	2	Mpiko Ntsekhe	needs to be inferred from dataset [1] by using identifiers	68	ID, demographic, clinical, haemodynamic, clinical laboratory, ECG and outcome data	xls	Redundant, some imputed data
IMPI phase 2 ECP outcomes	3	Mpiko Ntsekhe	needs to be inferred from other dataset as dates incorrect	68	ID, demographic, haemodynamic, HIV and outcome data	xls	Mac/ Windows date error, redundant
IMPI phase 2 ECP cytokines	4	Mpiko Ntsekhe	needs to be inferred from dataset [1] by using identifiers	61	ID, demographic, haemodynamic phenotype, cytokines	xls	Redundant
ELISA	5	Kerryn Matthews	needs to be inferred from dataset [1] by using identifiers	104	ID, cytokines	xls	3 redundant versions, no true source data, missing dates
ELISPOT	6	Kerryn Matthews	needs to be inferred from dataset [1] by using identifiers	41	ID, ELISPOT	xlsx	no true source data, missing dates

Name	ID	Investigator	Time period	N	Description of data types	Format	Challenges
FACS	7-22	Kerryn Matthews	needs to be inferred from dataset [1] by using identifiers	various	ID, numbers, proportions	xls,csv, pzf	16 files, no true source data, redundancy, missing dates
RT-PCR	23	Kerryn Matthews	needs to be inferred from dataset [1] by using identifiers	28	ID, delta CT	xlsx	Only some source data (SDS), missing dates
HIV-viral load	24-25	Kerryn Matthews	needs to be inferred from dataset [1] by using identifiers	16	ID, viral load PCF, HIV data	xls, xlsx	Missing dates
Whole blood assay	26	Kerryn Matthews	needs to be inferred from dataset [1] by using identifiers	59	ID, IFNγ responses to various antigens	xlsx	Missing dates
Immunology metadata	27	Kerryn Matthews	needs to be inferred from dataset [1] by using identifiers	110	Metadata of all immunology samples	xlsx	Missing dates
EU study clinical data	28	EU investigators (Raylene Titus, Tollulah Oni)	5/11/2007 - 26/8/2010	1263	ID, demographics, clinical, microbiology, imaging	File Maker DB	Missing data, sparse matrix, multiple sheets
EU study RNA data	29	Nzwaki Bangani	5/11/2007 - 26/8/2010	1233	ID, dates, RNA parameters	xls	
Prospective TB-PC cases	30	Armin Deffur	1/3/2011 - 2/2/2012	51	ID, demographics, clinical, imaging, ECG, echocardiography, clinical laboratory, microbiology, IMPI-MA sample metadata	xlsx	baseline data only

Table 3.3: (cont.) Sources of clinical phenotype data used in IMPI-MA

3.4.2 Integration strategy

The strategy for production of a single clinical phenotype database is outlined in Figure 3.6. This was a complex process requiring multiple rounds of data cleaning and development of computer code in *Mathematica* [113] that automated the process of TB class assignment based on the complex case definitions in Table 3.2.

1. Merge datasets 1-4; clean data and fully eliminate redundancy. Folder review of study and hospital folders to resolve any conflicts. Output = SET-1_clinical
2. Merge datasets 5-27; clean data and fully eliminate redundancy. Ensure correct matching of identifiers to individuals in clinical dataset. Output = SET-1_immunology
3. Merge SET-1_clinical and SET-1_immunology. Each row in flat-file database contains all data on on study subject. Output=SET-1_full
4. Identify potential microarray candidates using the following criteria: Blood RIN and PCF RIN of matched blood and PCF samples > 6.0
5. Export clinical subset of this data as SET-1 (immunology data removed as not available in SET-2 or controls)
6. Process SET-2: Ensure identical variable naming as SET-1 for integration purposes.
7. Entire SET-2 exported as microarray candidates as RIN data not available (RIN to be performed in UK)
8. In Mathematica, calculate tuberculosis class and export as SET-1 and SET-2
9. Merge SET-1 and SET-2 into full TB-PC clinical phenotype dataset
10. Search EU data for potential array candidates based on RNA availability and clinical phenotype (Healthy, LTBI and PTB)
11. Verify RNA availability for potential samples
12. Match the three groups to identify 20 each of Healthy, LTBI and PTB using age, sex, HIV-status, CD4 count as matching criteria
13. Extract additional relevant clinical phenotype data from EU database and export data as Controls, with identical column headers in the flat file datasheet
14. Manually merge TB-PC and Control datasets into single file
15. Duplicate phenotype entries as required, so that each sample type (blood and pericardial fluid) has a full phenotype record
16. Examine final dataset for missing data, and perform final folder review of various study and hospital folders to close gaps where possible
17. Export to production CSV file, to be used in all microarray analyses.

Figure 3.6: **Workflow for data integration**

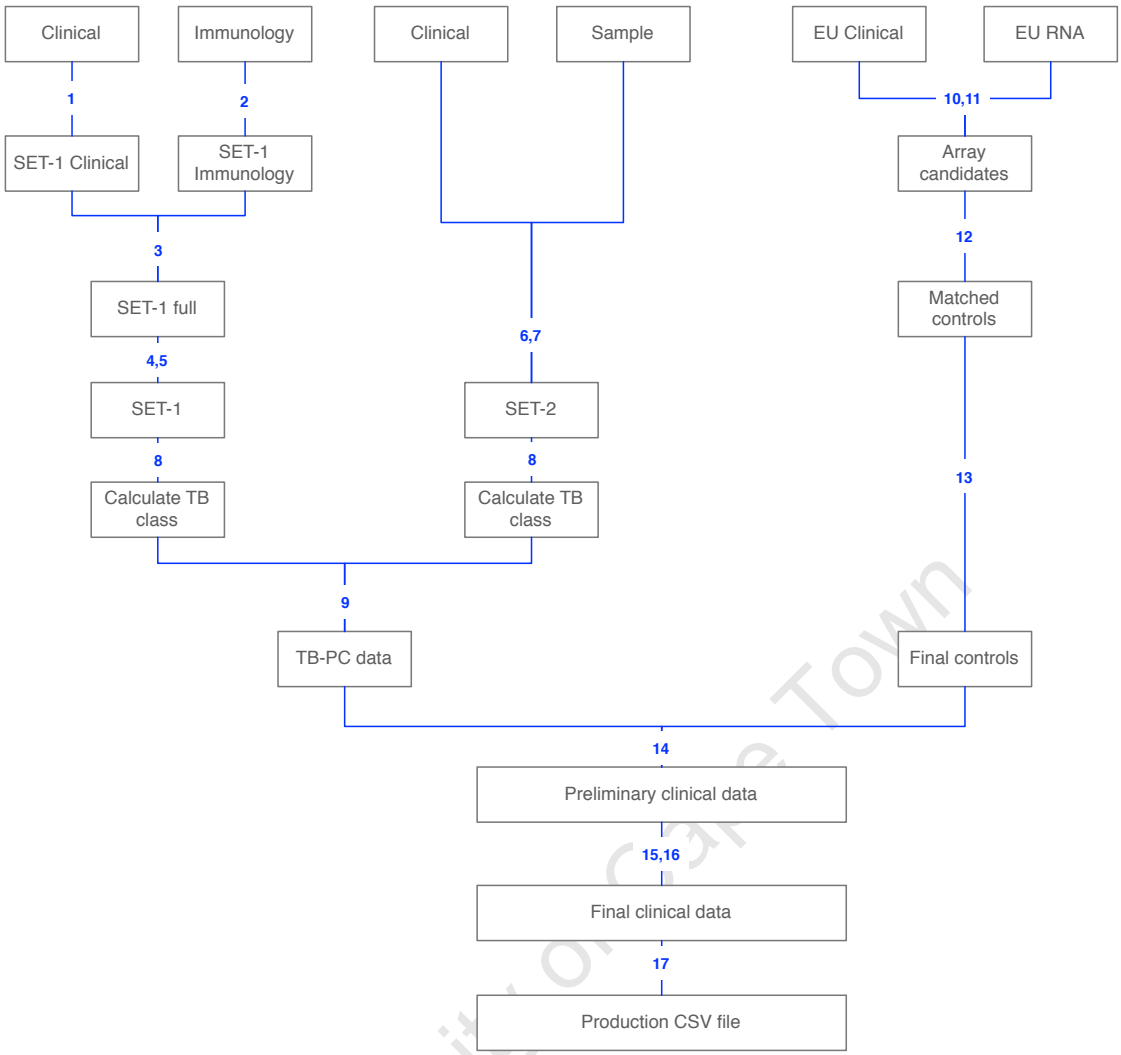


Figure 3.6: Workflow for data integration

3.5 RNA processing for microarray

3.5.1 Extraction and collation

RNA was extracted as described above, using the PAXgene Blood RNA Kit (PreAnalytiX, catalog number 762174) according to the manufacturer's instructions. RNA was extracted by different individuals and different times, depending on the original study for which the RNA was intended.

1. Retrospective TB-PC cases: Kerryn Matthews and myself

2. Retrospective controls (EU study): Nzwaki Bangani

3. Prospective TB-PC cases: Nzwaki Bangani and myself

As described previously, RNA samples were selected by different criteria depending on the original study. As all samples in the retrospective TB-PC case series had RIN data available, only high-quality samples were selected. The EU study controls had no RIN data available, so simply sample availability was used as criterion to select sixty microarray candidates. Lastly, the prospective TB-PC case series had no RIN data available, so all available matched blood and pericardial fluid samples were selected as array candidates.

3.5.2 Shipping and storage

The selected samples were shipped on dry ice to the National Institute for Medical Research in London (NIMR), UK, for further processing and microarray analysis, under an agreement with our collaborator there, Dr Anne O'Garra, head of Immunoregulation at the NIMR and senior author of the Nature paper [82] on which our laboratory had previously collaborated.

3.6 Microarray workflow

3.6.1 RNA preparation for microarray

All RNA processing for microarrays, as described in this subsection was performed by Christine Graham in the laboratory of Anne O'Garra, and her contribution to this project is hereby acknowledged. An outline of the process is shown in Figure 3.7.

The RNA preparation process followed the method outlined in [114]. The first step in preparing the extracted RNA samples for microarray was to verify RNA quality and yield, using an AB Bioanalyzer 2100 instrument at the NIMR. This step was used to select samples for further processing, as samples with degraded RNA (i.e. RIN values < 6) or low yield (i.e. yield <1250 ng) would not be expected to produce useful results. Selected samples were then subjected to a globin reduction step, in order to reduce interference from globin mRNA message with the transcripts of interest [115, 116]. Quality control was then performed on all globin-reduced samples, and all samples passing QC criteria taken forward the following step: cRNA generation. In short, the mRNA in the sample was converted to cDNA, which in turn was converted to cRNA. A final QC

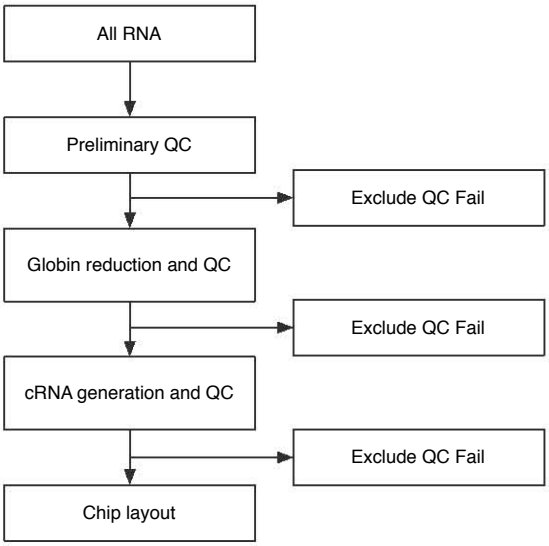


Figure 3.7: **RNA preparation schema**

step was performed to confirm high-quality cRNA for further use in microarray data generation. All samples were processed independent of original sample type (blood and pericardial fluid) and clinical phenotype. The samples passing all QC checkpoints were then randomly assigned to positions individual array chips. This array randomisation is vital to reduce the batch effects that are due to the specific chip used [117]. The workflow for the RNA preparation step is summarised in Figure 3.7.

3.7 Microarray data

3.7.1 Array technology

The BeadArray technology was first described in 2004 [118], as was a method used to decode such arrays [119]. All microarray data was generated on the Illumina BeachChip platform, using the current version of the assay and Illumina Human HT12-v4.0 BeadChips. Each BeadChip contains twelve arrays; generating multiple arrays on a single chip further reduces batch effects. Each array contains approximately 1-1.5 million beads, representing in excess of 47,000 probe sequences, randomly distributed across the array, with approximately 30 replicates of each probe type on each array. Given the random bead distribution, each array is unique, and requires decoding [119] prior to use. The decoded array is then used for RNA hybridisation.

1. Starting material: mRNA extracted from human sample
2. Globin reduction using GLOBINclear kit (Ambion, USA, catalog number AM1980) (a method utilising capture of unwanted RNA by magnetic beads)
3. cRNA generation and purification (Illumina TotalPrep -96 RNA Amplification Kit,) (Illumina, USA, catalog number AMIL1791)
    a) Reverse transcription to synthesise first-strand cDNA
    b) Second strand synthesis to generate double-stranded cDNA
    c) cDNA purification to remove RNA and other components that could potentially inhibit or interfere with downstream reactions
    d) *in vitro* transcription to generate multiple copies biotinylated cRNA from double-stranded cDNA templates
    e) cRNA purification to remove unincorporated NTPs, salts and other residuals which could interfere with the direct hybridisation assay

Figure 3.8: Workflow for RNA preparation

3.7.2 RNA hybridisation and addition of fluorophore

All microarray work, starting with biotinylated cRNA prepared by Christine Graham, was performed by Harsha Jani at the Microarray Core Facility at the NIMR. The workflow for the microarray work is listed in Figure 3.9.

1. Normalised quantities of cRNA are dispensed onto the BeadChip
2. The loaded BeadChips incubated overnight (10-14 hours) at 58°C
3. Wash BeadChips using high-temperature wash buffer and follow with room-temperature washes
4. Apply streptavidin-Cy3 to BeadChips; this will allow for signal detection using a fluorescent light source as the streptavidin-Cy3 fluorophore will bind to biotinylated cRNA present on the chip
5. Final wash steps and drying

Figure 3.9: **Workflow for array hybridisation**

3.7.3 Imaging

After hybridisation, the BeadChips were scanned at the Microarray core facility using an Illumina iScan microarray scanner. This step converted relative transcript abundance into a proportional light signal. This light signal was captured during the imaging step by a powerful CCD camera, which resulted in a high-resolution TIFF image. Each array was imaged in two passes, and the two resulting images were then combined into a single image by the iScan software. At this stage, the chip decoding data, unique to each BeadChip was loaded onto the controlling computer, in order to assign bead identifiers to each bead on the chip.

3.7.4 Image processing and data summarisation for further analysis

The decoded high-resolution images generated by the iScan scanner were processed further using GenomeStudio, resulting in summarised probe-level data. Each step in the summary process reduced the number of observations per array. This is illustrated for a single array in Table 3.4.

Table 3.4: **Levels of data complexity**

Level	Data type	Interpretation	Number of observations
Raw image	Pixel intensities	One observation per pixel per array	approx. 125×10^6
Bead-level	Bead intensities	One observation per bead per array	approx. $1.0 - 1.5 \times 10^6$
Probe level	Bead intensities summarised by bead type	One observation per probe type per array	47,231

The summarised bead intensity value consists of an average intensity value for a particular bead type, together with the number of replicates and the standard deviation for the observation. This is useful for the data transformation step.

The final microarray data was provided by the NIMR Microarray Core Facility as a tab-delimited text file containing summarised probe-level data.

3.8 Data analysis overview

3.8.1 Computing environment

All data analysis was performed on a mid 2010 MacBook Pro with dual-core Intel Core i7 processors and 8GB non-ECC RAM running OS X 10.8.4. R software was used for the entire analysis workflow.

3.8.2 Analysis framework

All data analysis was done in R version 2.15.2 using a set of scripts, and utilising several R and bioconductor packages.

General data analysis revolves around analysis of RT-PCR data (see above), presentation of an unbiased view of all microarray data, and demonstration of the relation of the IMPI-MA data to previous studies of tuberculosis using blood transcriptomics approaches.

The core of the analysis revolves around seven questions (see Table 8.1) pertaining to the three main hypotheses. Multiple analyses of different, overlapping subsets of the data in fact constitutes a meta-analysis, as the goal is not only to evaluate a single contrast and reach a conclusion, but to evaluate multiple contrasts, compare the results and then reach a more generalised conclusion about the pathogenesis of HIV-TB in a two-compartment model. Each of the seven questions is addressed using complementary analytic methods: detection of differential gene expression for the contrast of interest, deconvolution of the data in order to estimate proportions of specific cell types in the sample as well as cell-type-specific differential gene expression, and modular analysis by detection of modules on the basis of co-expression networks in the microarray data. The scientific basis and methodology underlying these analyses will be described in the remainder of this section. The remaining sections of this chapter, together with Chapter 8, outline the R scripts that underlie various aspects of the analysis, and shows how these scripts together constitute an end-to-end solution for reproducible and well-documented (meta-) analysis of heterogenous microarray datasets that consist of data based on samples from different tissue types and individuals with different disease phenotype classes. All source code is reproduced with syntax highlighting in the Appendix (A), and has been annotated to facilitate readability. The scripts produce output in three main classes: RData files for further processing, images and delimited text files that can be imported in other

applications (e.g. Microsoft Excel, Cytoscape).

3.8.3 Microarray analysis in general

The approach to studies involving microarray data should include pre-defined research questions. Overall hypotheses must be stated prior to the analysis of the data, and the results presented in the context of the hypotheses. Pre-stating clear hypotheses facilitates the process of selecting a data analysis goal and therefore tools to achieve this goal. In general supervised and unsupervised methods may be applied to microarray data, and their use depends on the research question (see below).

Analysis of microarray data usually has one or more of three goals [120]: Class Comparison, Class Prediction and Class Discovery. The aim of *class comparison* is to find genes that are differentially expressed between groups of samples that belong to known and defined classes. The analysis is supervised, as the classes are known and this knowledge is used to find differentially expressed genes. *Class prediction* uses unsupervised methods to separate classes in the data without prior knowledge. These predicted classes can then be compared to the real classes, and sensitivity and specificity calculated. Usually, the classifier is built using a training set, and then tested on a separate test set. *Class discovery* again uses unsupervised methods to discover previously unknown classes of samples. This may be used where sample phenotyping is unable to further differentiate phenotypes within a set of samples that is suspected to be heterogenous at the molecular level. This approach has been used in oncology to classify morphologically identical tumours into subclasses that can be linked to survival or other clinico-pathological outcomes [121, 122].

3.8.4 Reduction in technical bias: array transformation and normalisation

As stated previously, there are many factors (which are not due to the controlled factors in an experiment) that may influence the intensity of a particular probe type on a BeadArray chip, for instance quantity of mRNA. In order to reduce the impact of this source of bias, array data should be normalised. Multiple methods as applied to Illumina HumanHT12 v3 BeadArray were compared in a recent paper[123], and on the basis of that paper, as well as recommendations in the documentation of Illumina-specific analysis packages in R [124], variance stabilising transformation followed by

quantile normalisation was performed.

3.8.5 Differential expression analysis

The most fundamental analytic technique applied to microarray datasets is the detection of differential expression of probes or genes between two or more classes of samples. While simple in principle, in practice this problem is quite complex. The individual measured intensity values are affected by multiple sources of fluctuation and noise. This problem is in part addressed by including technical replicates on the microarray chip; the Illumina HumanHT12v4 chip used in this study contains an average of 30 replicates for each probe type on the chip.

Multiple methods for detection of differentially expressed genes have been described. Many rely on some form of statistical hypothesis testing, where the null hypothesis $\mathfrak{H}_o$ is that there is no difference in transcript abundance for probe (or gene) x. A simple method of testing this hypothesis is to perform a simple t-test; all genes that are significant at a pre-specified p-value are labelled as differentially expressed. This approach is likely to yield many false-positives, given the typical number of tests performed (20,000-48,000 for Illumina data). A more rigorous approach would be to subject the results to some form of multiple testing correction. A non-exhaustive list of potential methods for selecting differentially expressed genes is shown in Table 3.5 (Table based on [125]).

In the analysis of the IMPI-MA data, we employ an approach first described by Smyth [109], namely the use of a moderated t-statistic in selecting differentially expressed genes. This method is widely used in the literature, and yields robust results.

This approach aims to fit a linear model to expression data that fully models the systematic part of the data. The model is specified by a design matrix, where each row represents one array, and each column represents one coefficient that describes the RNA source (i.e. a phenotype variable). The purpose of this approach is to describe and account for the variability of the data. For single channel arrays, linear modelling is similar to ordinary ANOVA or multiple regression, except that a model is fitted to every gene (or probe type). Once the model is derived, an empirical Bayes method is used to moderate the standard errors of the estimated log-fold changes. This results in estimators that have robust behaviour, even for small numbers of arrays. The posterior odds statistic is thus reformulated in terms of the moderated t-statistic where the posterior residual standard deviations

Table 3.5: **Methods for selecting differentially expressed genes**

Name	Method	Advantages	Disadvantages
Fold Change	Select threshold	Simple and intuitive	arbitrary
Unusual Ratio	Ratio of experiment to control > 2 standard deviations from mean experiment/control ratio	Simple	always selects 5% of all genes
Hypothesis testing	t-test followed by multiple testing correction	Powerful	Very conservative, assumes genes are independent
ANOVA	Explicit model of all sources of variance	Each source of variance is accounted for	Requires specific experimental design
Noise sampling	Estimate noise and calculate confidence levels for gene regulation	Better sensitivity than unusual ratio and better specificity than fold change methods	Provides estimates only for log-ratios of genes
Model-based maximum likelihood estimation	Generate mixture model for distribution of observed log ratios for expressed and non-expressed genes	General, powerful, flexible	Require solution of complex non-linear equations; unreliable for smaller samples
SAM	Permutation-based to discover genes differentially regulated for a given false discovery rate	Useful for small sample sizes	
Moderated t-statistic	Fit linear model to each gene and determine whether coefficient differs significantly from zero	Useful for small sample sizes; powerful	

are used in place of ordinary standard deviations, resulting in more stable inference.

In the model, the array intensity values are represented by a continuous variable termed the *response variable*. The contrast(s) of interest are represented by one or more categorical variables (*explanatory variables* or *predictors*). The goal of the procedure is to test the the association between predictors and response. Equation 3.1 shows the model:

$$y_i = \sum_{j=1}^{p} \beta_j x_{ij} + \beta_0 + \varepsilon_i \tag{3.1}$$

Here, y_i is the response for the ith sample, β_j the coefficient for the jth predictor, x_{ij} the value of the predictor x_j of the ith sample, β_0 the intercept of the model, and ε_i the error term (or residual), representing noise and the effect of all predictors not considered in the model.

A microarray experiment consists of a large number of probes or genes that have to be modelled, therefore the equation above can be represented in matrix form as

$$\mathbf{y} = \mathbf{X} \cdot \beta + \varepsilon \tag{3.2}$$

where $\mathbf{y}$ is the response vector, $\mathbf{X}$ the design matrix, β the vector of coefficients and ε the vector of residuals. Building a linear model then reduces to finding the values of the βs (coefficients) that would minimise the error, i.e. solving $y = X \cdot \beta$. The R package *limma* implements an ordinary least squares solution, as $\mathbf{X}$ is usually not a singular square matrix, and makes the following assumptions:

1. The residuals ε_i are normally distributed
2. The mean (expected value) of the residuals is zero and is independent of the values of the predictors
3. The variance of the residuals does not depend on the values of the predictors (homoscedasticity)
4. The value of the residual for one sample is unrelated to the amount of residual for another sample

A basic limitation of parametric modelling is that this approach only finds the best values for the parameters available in the model.

Once the fitted model with the moderated t-statistic is produced, multiple testing correction is applied in order to reduce the number of false-positive calls. Throughout all analyses, a multiple testing correction method described by Benjamini and Hochberg [126, 127], based on a specified false discovery rate has been applied.

Issues with this study:

This study aims to compare the results of multiple differential expression analyses, each based on a subset of data with an variable number of samples in each contrast category. As each differential expression analysis attempts to identify an effect size that may differ widely between sample subsets, from small (difference between healthy and LTBI in blood in HIV uninfected) vs. large (e.g. differences between gene expression in matched blood and pericardial fluid samples), the statistical power of each DE analysis is different. Some sample subsets consist of large numbers in each group, and the likely effect size is large, while others consist of small numbers and aim to detect a small effect size, therefore the type II error rate for each DE analysis is different.

This is dealt with by limiting the number of called DE probes to a maximum of 2000, and by including as a minimum the top 300 DE probes, regardless of whether the adjusted p-value is significant or not. In the latter case, such an analysis will be regarded as exploratory, but may still offer some biological insight and aid in the development of testable hypotheses.

3.8.6 Deconvolution and cell-specific differential expression

A common criticism of microarray approaches in general is that the expression profiles for a given sample strongly depend on the cellular composition of the samples in question. Now, if the cellular composition between samples varies widely, differences in gene expression profiles are in the first instance ascribable to the differences in cellular composition, and not necessarily to differential use of transcriptional pathways. In addition, the power to detect differentially expressed genes between samples is strongly affected by sample variation in cell-type frequencies [128, 129, 130]. Therefore, comparisons of gene expression between samples of different tissues (like blood and pericardial fluid) are likely to yield many differentially expressed genes, most of which are uninformative regarding true biological differences, and subtle differences in cell-type specific gene expression in closely related conditions may be masked by the noise of the cell-type frequency component in the

expression signal. To solve this dilemma, two approaches can be taken.

First, the samples may be separated into constituent cell populations. For example, a whole blood sample may be separated into pre-defined cell subsets (neutrophils, CD4 T cells, CD8 T cells, and others). Then, RNA is extracted separately and microarrays performed. This approach has a number of disadvantages: it is time-consuming and expensive, must be done prospectively, and the physical separation of the cells may have unexpected effects on gene expression in the separated cells [131, 132]. This approach was not followed in this study.

The second option is to determine the proportions of cell types in the samples, and then perform a cell-type specific differential expression analysis on the expression matrix based on these proportions. Determination of cell-type proportions in each sample may be performed by physical means (e.g. flow cytometry) or computationally, using a method called matrix deconvolution and first described by Lu et al [133]. The analysis of the IMPI-MA dataset implements the matrix deconvolution approach utilising a least-squares fit algorithm described by Abbas et al [100] and implemented in the *CellMix* package [134].

Estimation of cell-type specific differential gene expression may also be performed computationally. This relies on known (or estimated) proportions of cells for each sample, and subsequent application of differential expression analysis. A method utilising the permutation-based method called *significance analysis of microarrays* (SAM) has been described [135] and will be used in this analysis. This method is also implemented in the *CellMix* R package. Estimates of cell-type specific differential expression may uncover true biological differences between samples with variable cellular composition.

A number of methods for gene expression deconvolution have been described (e.g. [100, 136, 137, 138]) and a full review can be found in the comprehensive documentation of the *CellMix* package. Here I briefly describe the method used by Abbas et al [100], as this method was most suitable to the data to be analysed. The objective of the deconvolution method is to find the solution for a convolution equation

$$\mathbf{AX} = \mathbf{B} \tag{3.3}$$

Here, $\mathbf{B}$ is the matrix of gene expression data, $\mathbf{X}$ is the matrix of proportions of cell types in all

samples and **A** is the matrix of known expression levels of genes in all cellular components of **B**, which is convolved with **X**. This equation is solved for **X** by standard least squares fitting using the method pioneered by Lu [133], where one solves for **X** with the constraint that the resulting matrix is contains only non-negative entries.

This approach requires knowledge of **A**, and in the Abbas paper [100], the authors describe their method of generating this matrix of cell-type specific signatures, its validation and subsequent application to a whole-blood microarray study of systemic lupus erythomatosus. The *CellMix* package implements multiple deconvolution methods. One method (*gedBlood*) replicates the Abbas method exactly, and utilises the above cell-type signature matrix; this was used in the analysis of the IMPI-MA data.

Once cellular proportions for all samples represented in the expression matrix are known, differential gene expression for each cell type represented in significant numbers should become possible. A method to achieve this was published in 2010 [135].

The following two assumptions are made in the analysis as reported:

1. Blood and pericardial fluid are assumed to have the same cellular composition, differing only in proportion

2. These cell-types consist of the cell-types isolated and studied by Abbas

In a study published in 2006 [139], Reuter et al performed flow cytometry on whole blood and pericardial fluid in patients with tuberculous pericarditis (including HIV-1 infected and -uninfected groups). Their findings show that the three most commonly cell-types were neutrophils, lymphocytes and monocytes, while NK cells also occurred. The data also show that proportions of cell types in blood and pericardial fluid, in both HIV-1 infected and -uninfected individuals, are markedly different, with neutrophils being the most abundant cell-type in blood, but much less frequent in pericardial fluid.

Despite the promise of using microarray data in order to infer cell proportions, some limitations of this approach need to be recognised. Firstly, this approach only yields indirect evidence of different cell proportions and cell-specific differential expression, and the results require verification prior to accepting the results at face value. An experiment that does this is described below. Secondly, the ability to detect differences in whole blood may be limited due to the dilution of the

signal in mixed cell populations. In published papers[100, 135] however, the authors are able to reliably detect cell proportions and cell-specific differential expression. Indeed, sample heterogeneity may mask true differential gene expression when performing differential gene expression analysis on whole blood transcriptomes, while cell-specific differential expression correctly identifies the cell type and transcript list of differentially expressed genes[135].

Validation of the deconvolution approach Given the limitations discussed above, validation of the deconvolution method is warranted. This would be best achieved by collecting flow cytometry data on whole blood on the same samples on which array data is available. In the absence of such data (no whole blood flow cytometry data was generated for IMPI-MA), I used existing data generated in the Berry et al study [82] (London Test Set) to partially validate this method.

Flow cytometry data was generated on whole blood samples on 27 patients, approximately equally sampled from healthy controls, LTBI and PTB cases. 17 of these patients also had Illumina HT12v3 microarray data available. All flow cytometry data was provided by Fin McNab in the Anne O'Garra laboratory, NIMR, London, UK.

The flow cytometry data was received as three Microsoft Excel files; one each for healthy controls, LTBI controls and PTB cases. For each workbook, one worksheet was dedicated to one class of cells. I copied the original data into a new Excel workbook with one worksheet for each class of cell, converting all cells containing formulas to static values; each worksheet contained data for all patients. Next, I removed all rows (samples) from this for which no microarray data was available, and all columns which did not contain relevant data for the intended comparison.

The following cell subsets were enumerated: Total T cells, CD4 T cells (and multiple subsets), CD8 T cells (and multiple subsets), other T cells (i.e. double negative for CD4 and CD8), dendritic cells (plasmacytoid and myeloid subtypes), inflammatory and non-inflammatory monocytes, NK cells, and neutrophils. Full methods describing the flow cytometry approach are available in the supplementary data for the Berry et al paper [82]. Based on proportions seen at flow cytometry and total number of white blood cells as based on complete blood counts, numbers of each subtype were calculated. The sum of numbers for all cell types was computed, and compared to the total white blood cell count; this yielded the number (and proportion) of cells unaccounted for by flow cytometry for the particular sample. Finally, the subset of the data representing the cell propor-

tions matrix similar to the proportions matrix yielded by deconvolution was exported as a comma delimited text file and used for further analysis. Only samples where the sum of all proportions represented at least 80% of all cells were included. This aims to exclude noise introduced by additional cell types not measured by flow cytometry but which may influence the expression profiles derived from whole blood.

Deconvolution using the method described above was applied to the full BERRY test set. Cell types identified by deconvolution included CD4 T cells (resting and activated). CD8 T cells (resting and activated), B cells (5 subtypes), plasma cells, monocytes (resting and activated), NK cells (resting and activated), dendritic cells (resting and activated) and neutrophils.

In the analysis step, the two proportions matrices obtained from flow cytometry and deconvolution, using the same samples and containing cell subsets with identical labels, were used. Figure 3.10 shows how the cell subsets from the two approaches were related to each other to study a unified subset of common cell types.

The function *profplot*, implemented in the *CellMix* package, was used to plot the combined proportions matrices. For each cell type, the regression coefficient (α), Spearman rho (ρ) and Pearson R^2 with 95% confidence interval were calculated. The results of the analysis are presented in Chapter 5.

3.8.7 Weighted gene co-expression network analysis (WGCNA)

The techniques described above for differential expression focus on finding individual genes that may be important in a particular biological context. However, as described in Chapter 1, the focus of systems approaches is to find explanations for behaviour and properties of complex systems, i.e. identification of functional units instead of individual parts. Modules of co-expressed genes may be related to emergent and other high-order phenomena of such complex systems, leading to improved understanding of the underlying biology [140]. This leads to a reduction in high-dimensional data, and allows for integration of multi-scale data.

An implementation of such an approach was first described in 2005 [140] and has been implemented as the package *WGCNA* in R [141].

The fundamental concept underlying the functionality of the *WGCNA* package is that genes are

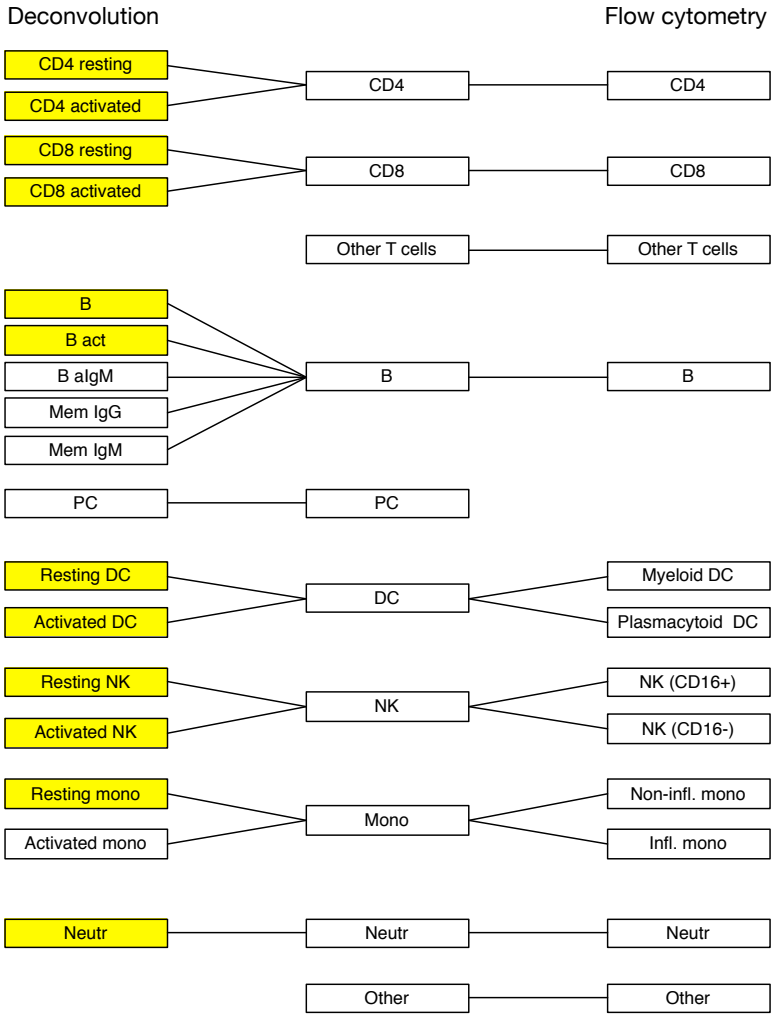


Figure 3.10: **Approach for unification of cell type subsets** For the deconvolution cell subsets, those subsets actually detected by the deconvolution algorithm in the array data are marked in yellow.

expressed in groups called modules, and that such modules are identifiable in microarray expression data. The processes involved in detecting such modules and interpreting their function and importance relative to phenotypic or other information is described below.

The network is constructed by making use of interaction patterns in the raw expression data. Pearson correlation is used as the measure of describing such interaction, as it is intuitive, avoids overfitting, is computationally fast and is reproducible.

The first step in the analysis is therefore construction of the correlation matrix, containing the Pearson correlation coefficient for each pairwise interaction for all genes studied. For large gene lists this is a computationally intensive step, and the package provides some utilities to run the analysis on systems with relatively small amounts of RAM. For instance, a list of 8000 probes or genes generates a square matrix with sixteen million entries. The next step of the analysis transforms the correlation matrix into an adjacency matrix, where each entry represents the weighted connectivity between all pairwise gene interactions, by using the power function

$$f(x) = x^{\beta} \tag{3.4}$$

The parameter β is chosen in such a way that the resulting adjacency matrix exhibits scale-free topology, as many biological networks that have been studied experimentally exhibit scale-free topology [142, 143, 144, 145, 146, 147]. This is based on a previously described model fitting index R^2 [140]. For a perfectly scale-free network, this index is defined to equal one. Here, the parameter β is chosen as the smallest exponent of the above power function such that the model fitting index $R^2 \geq 0.8$. This is termed the soft-thresholding procedure. It preserves the continuous information of the co-expression network and yields robust results with regard to different threshold choices [148]. The result of this step is an adjacency matrix, where each entry corresponds to the strength of the interaction between the two genes. This adjacency matrix is then transformed into a topological overlap matrix (TOM) which enables rapid detection of modules. This matrix provides a similarity measure for each pairwise interaction reflecting their relative interconnectedness.

Detection of modules (i.e. groups of genes whose expression profiles are highly correlated across samples) is performed as the next step. First, the TOM-based dissimilarity matrix (i.e. 1-TOM) is clustered using average linkage hierarchical clustering. Modules are then detected based on the

clustering result, and this therefore depends on the choice of branch cut height, i.e. how distant two clusters have to be to belong to separate modules. The WGCNA package [141] implements a dynamic branch-cutting algorithm that allows for variation in branch cut height for different modules.

In order to interpret the modules, they are next related to external information. For the purposes of this study I have limited this to the contrast of interest, in order to identify modules important in that contrast. This allows for the discovery of "interesting" modules that may be analysed further. Module eigengene expression can be linked to clinical data. This is implemented by correlating eigengene expression values with clinical trait data. As clinical traits can be measured using continuous or categorical variables, this presents us with a problem. To address this, categorical clinical traits were first converted to ordinal data by encoding the individual categories using numerical values either arbitrarily (e.g. sex), or where appropriate on scale of increasing severity (e.g. healthy, latent TB, active TB). This allows for associating module eigengene expression with both continuous variables as well as categorical variables.

Finally, module preservation across different datasets can be studied, and the key drivers (hub genes) in the modules identified.

3.8.8 Pathway analysis

Exploratory pathway analysis was performed on several datasets, as proof of concept and for future integration in the analytic framework. Pathway analysis was performed using the R package *gage* [149], and pathways were visualised using the R package *pathview* [150]. Since these packages require R version 3.0.1 or higher, the code was not integrated with the other scripts, as the current analytic framework still runs in R version 2.15.2; code integration will require updating the R installation to R version 3.0.1 and all Bioconductor packages to version 12.

Gage implements pathway analysis using publicly available data from KEGG (Kyoto Encyclopedia of Genes and Genomes) [151, 152]. The method extends techniques used for gene set enrichment analysis and provides a more robust environment for finding differentially regulated pathways. It allows for unpaired samples, and case and control groups of different sizes. *Gage* outputs over-represented pathways ranked by q-values.

All pathways significant after multiple testing correction were visualised in using *pathview*, with log-fold change values mapped to individual pathway components, indicating whether this component was over- or underexpressed in the differential expression analysis.

Pathway analysis has several limitations. These have been reviewed recently in [153] and are summarised here. These limitations can be divided into two main groups: Annotation challenges and methodological challenges.

Annotation challenges Genomic annotation databases at present lack the resolution of new-generation technologies used to generate state-of-the-art genomic data. Alternative splicing may, for example, be detected using RNA-seq, but annotation databases may contain only information for a single transcript. Multiple transcripts of the same gene may have different, even opposing, biological functions, but this is not reflected in current databases. So, a query to KEGG will only indicate that a given gene shows altered transcript abundance, but does not identify the biological function of the specific splice variant. In addition, annotations contained in biological knowledge-bases may be incomplete or inaccurate. Many genes are annotated as "hypothetical" genes with no known function. Despite this, transcripts that map to these non-annotated genes appear to be important in biological contexts. Finally, information on specific pathways is usually based on a defined experimental system, but information regarding the experimental context is often missing from pathway resources.

Methodological challenges Benchmark data for comparing different pathway analysis methods is not available. Methods are usually compared using simulated data, which yields reproducible results but no indication of which method correctly identifies real biological phenomena. Also, pathways are not independent of each other, yet analytical methods usually make that very assumption. This results in a lack of a model that accounts for dependence among pathways at different time points. This limits our ability to observe changes at a pathway level in a biological system. This may in part be addressed by topology-based approaches.

Conclusion Pathway analysis, as implemented for this thesis has several limitations. Care must be taken when interpreting results.

3.8.9 Other methods

Multiple additional methods to analyse microarray data exist. I chose the above to provide a broad view of the transcriptomic landscape in HIV-TB across many different contrasts. An additional method that might have been implemented is transcription factor binding site analysis. This would yield information regarding possible regulatory factors which might explain the observed variation, and would definitely improve the biological insight gained from the various gene lists. Future work may add this and other methods to the pipeline described above.

3.9 Code descriptions

In this section, the R code used for the analyses is described in outline. Full code listings with syntax highlighting are provided in the Appendix. Additional information on customised R functions used in the setup and analysis scripts can also be found in the appendix, together with the code for the functions.

Some of the principles that are used in the code, specifically the idea of setting up the workspace environment, as well as the general workflow for the the code in subsection A.9 are based on a tutorial for performing the analysis of the Berry et al data, reported in 2010 [82] in R instead of GeneSpring. I adapted the code as required; the utility functions mentioned on the website have been incorporated in the system setup script.

IMPI-MA refers to data generated for this PhD project; BERRY refers to expression data published in 2010 [82].

3.9.1 IMPI-MA: System setup

The code for this analysis is reproduced in full in Appendix Section A.6. This script is sourced by all other analysis scripts, and standardises the analysis and output environment for each analysis. The script performs the following actions:

1. Initialise the session by clearing the workspace and resetting the graphics devices
2. Configure local folders (project root, R-scripts folder, data folder)

3. Load utility functions
4. Set global parameters (e.g. image export formats, graphics parameters)
5. Load all libraries required by the analysis scripts

3.9.2 IMPI-MA: Data manager

The code for this analysis is reproduced in full in Appendix Section A.7. This script performs the following actions:

1. Source system setup script
2. Specify local data and output folders
3. Import expression (IMPI-MA and BERRY) and phenotype data using lumi
4. Make subsets of the datasets using CSC (compartment, subset, contrast) scheme
5. Export datasets to RData files; any subsequent use of the data is greatly sped up using RData
6. Import 393 and 86 transcript lists, and export for later use as RData files

3.9.3 RT-PCR data analysis as a prototype for more complex analyses

The code for this analysis is reproduced in full in Appendix Section A.8. The script performs the following actions:

1. Setup
2. Source setup script
   a) Make specific data and versioned output folders
   b) Define additional functions *redgreen* and *interleave*

3. Read in RT-PCR and phenotype data and make single data structure
4. Calculate statistics for summary table of participant clinical characteristics
5. Select phenotype categories used for colour in plots
6. Make data matrix used in DE analysis and matrix of relevant phenotype information
7. Plot raw data: box-and-whisker plots, heatmap and correlation matrix plots
8. Filter data based on low expression values
9. Plot filtered data
10. Differential expression analysis for contrast: compartment
11. Plot and export results
12. Principal component analysis of all, filtered data and selected genes
13. Protein analysis
    a) Select protein data
    b) Boxplots and statistics for differential protein abundance
14. Additional analysis for *HIV status* and *lowCD4* contrasts in blood and pericardial fluid

3.9.4 Relation of IMPI-MA data to previously published work

The code for this analysis is reproduced in full in Appendix Section A.5. The code for the differential expression analysis (Baylor Method) is based on concepts described in an online tutorial (found here: http://www.bigre.ulb.ac.be/courses/statistics_bioinformatics/practicals/microarrays_berry_2010/) and adapted for use in this framework, but the remainder is original work.

The script performs the following actions:

1. System setup
2. Source setup script

a) Create local data folders and versioned output folders

3. Load data and place in a list which is iterated over for each step of the analysis
4. Average normalisation (replicating a procedure originally carried out in BeadStudio)
5. Calculations for two-way table of clinical characteristics, comparing variables between datasets, and between different disease classes within datasets
6. Show raw data using box-and-whisker charts, multidimensional scaling and principal component analysis
7. Data preprocessing: threshold low expression values to minimum of 10 and perform median scaling; visualise data
8. Data filtering: detection filter and fold-change filter followed by selection of probes passing both filtering criteria; visualise data
9. Calculate differential expression statistics
10. Generate lists of differentially expressed probes
11. Plot results and export lists of probes for further analysis

The code for the main analysis will be discussed in the Chapter 8, as development of this code depended on specific technical results; the overall code structure was conceived only after the microarray data had become available, together with information regarding the numbers of samples in various classes with successful arrays.

3.9.5 An unbiased view of all IMPI-MA and BERRY data

The code for this analysis is reproduced in full in Appendix Section A.10. This simple linear script performs the following actions:

1. Source setup script
2. Create local data and versioned output folders
3. Load data (IMPI-MA and BERRY)

4. Generate quality control plots, including density plots, boxplots and MA plots

5. Multidimensional scaling analysis of IMPI-MA data (all samples, blood and fluid)

3.10 Comparisons

Comparisons of gene lists were made by calculating the intersection of two or more gene lists, and visualising these using Venn diagrams in R; Code examples are shown in Appendix Section A.9.

Part II

General Results

4 RT-PCR on matched blood and fluid

Chapter summary

This chapter describes the results of an analysis of RT-PCR data from patients with tuberculous pericarditis, with particular focus on contrasting gene expression between the blood and pericardial fluid compartments. We show that significant differential expression of genes is detectable between compartments, and that this difference may be biologically informative.

Disclosure: The text is largely based on a manuscript in the early stages of preparation. All text was written by myself, but incorporates significant input from the senior author of the manuscript, Prof. Katalin A Wilkinson. For this reason, this chapter is written in the first person plural form. As described in Section3.1, all data was generated by Kerryn Matthews and the IMPI Registry team.

4.1 General comments regarding the presentation of results

Throughout this thesis results of increased or decreased transcript abundance are reported, be it generated by RT-PCR or microarray. The terms “upregulation” and “downregulation” are frequently employed to refer to increased and decreased transcript abundance as a form of convenient shorthand. In reality, the observed changes are usually due to factors over and above changes in the regulation of expression, and the use of the terms up- or downregulation should be read in their (intended) wider sense.

Many images depicting clustered versions or PCA plots of pre-selected features are shown in the results. These heatmaps and plots essentially show only that the selected features classify the input data correctly. Future work will include experiments where the signatures obtained from the IMPI-MA data will be validated against unrelated datasets.

Interpretation of PCA plots can be rather subjective. As it stands, the current work does not provide more objective metrics of similarity of datasets based on visual inspection of the scatterplots. Future work will incorporate calculation of Euclidean distances between centroids and drawing ellipses. R packages that have these features exist.

4.2 Study subjects

27 patients with TB pericarditis (15 definite and 12 probable cases) were included in this analysis (Table 4.1). Ten were HIV-1 uninfected and 17 were HIV-1 infected, with a median CD4 count of 167.0 (98.0-343.0) cells/μl. There was no difference in proportions of definite or probable TB diagnosis, gender or age between HIV status groups, but the CD4 count was significantly lower in the HIV-1 infected patients (p=0.025). One HIV-1-infected participant was on antiretroviral therapy at the time of sampling; duration of antiretroviral therapy was not recorded.

Table 4.1: **Clinical characteristics of study participants.** Proportions were compared using chi-squared test, and medians of continuous variables (age, CD4 count, pericardial fluid ADA and IFN)γ were compared using Wilcoxon rank sum test (Mann-Whitney U test)

Comparison of clinical characteristics of patients with tuberculous pericarditis, stratified by HIV-status				
	All participants (N=27)	HIV-1 uninfected (N=10)	HIV-1 infected (N=17)	P-value
Age Median (IQR)	31.6 (28.7-41.3)	47.4 (28.9-68.2)	31.4 (28.9-35.8)	0.155
Gender Female: Male	10:17	3:7	7:10	0.866
TB status Probable: Definite	12:15	6:4	6:11	0.397
CD4 count (cells/µl)	167.0 (98.0-343.0)	306.5 (190-418.5)	118 (89-266)	0.025
Concurrent steroids Yes:no	18:8	5:4	13:4	0.230
Fluid ADA (U/L)	77.0 (54.5-91.5)	81.0 (65.0-137.0)	75.0 (52.0-86.0)	0.374
Fluid IFN-ϒ (pg/ml)	1531.0 (205.5-4212.0)	2084.25 (0.0-3993.0)	1308.0 (498.0-4512.0)	0.907
Number on ART (yes:no)	NA	NA	1:16	NA

4.3 Gene expression analysis

Overall expression values in the two compartments were similar (Figure 4.1).

As a first step in our analysis, we grouped the 42 genes studied into eight functional categories: Th1, Th2, TH17, genes associated with immunoregulation, neutrophil associated, growth factors, fibrosis associated and matrix metalloproteinase associated genes. This categorised gene expres-

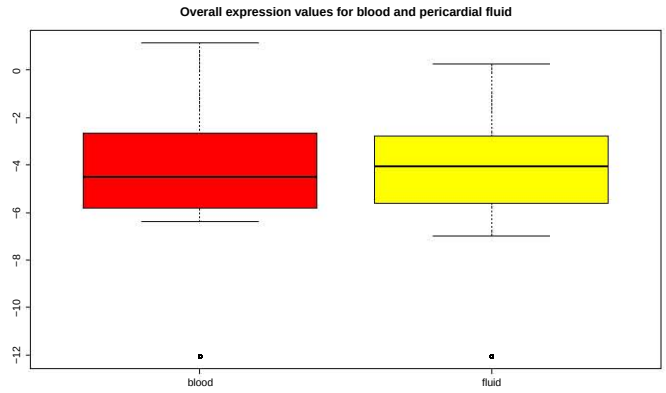


Figure 4.1: **Overall expression values.** Box-and-whisker plot showing the median, upper and lower quartile of raw ΔCT values for blood (red) and pericardial fluid samples (yellow)

sion in blood and pericardial fluid is summarised in Figure 4.2. Gene-by-gene comparison (with Benjamini-Hochberg multiple testing correction) yielded 21 genes that are differentially expressed in blood compared to pericardial fluid following application of Benjamini-Hochberg multiple testing correction (Table 4.2).

Overall, differences between compartments for the expression of Th1, Th2 and Th17- associated genes were not evident, whereas genes associated with immunoregulation did show a difference. In addition, overall differences in expression levels of fibrosis-associated genes were evident, as well as some neutrophil-associated genes, growth factors and MMP genes, as well as their regulators.

As a second step, we employed cluster analysis in order to explore the data further. By combining the expression values of all samples and all genes (as described in methods) we generated a heatmap shown in Figure 4.3. While there was no differentiation between blood and fluid, we found that strikingly, most blood and pericardial fluid samples from individual patients clustered together. This appeared to be driven by highly correlated gene expression patterns between the two compartments, specific for each individual, thus obscuring differences of gene expression between the two compartments. Four clusters of genes can be seen based on very low, low, intermediate and high expression regardless of compartment. Many genes appear to be expressed in groups. This phenomenon of co-expression was further explored in these samples and is shown in the correlation matrices of blood and pericardial fluid samples in Figure 4.4. Gene co-expression patterns in blood were different from pericardial fluid, where pronounced co-expression of fibrosis-associated

Table 4.2: **Gene-by-gene comparison of expression between compartments.** All comparisons statistically significant (P≤0.05) are highlighted in yellow.

Gene	Pval	Bonf	BH
IL1B	0.00	0.02	0.00
IL2	0.18	1.00	0.30
IL6	0.00	0.00	0.00
IL18	0.98	1.00	1.00
IL24	0.65	1.00	0.81
TNF	0.17	1.00	0.29
IFNG	0.02	0.63	0.03
CXCL10	0.30	1.00	0.45
IL4	0.05	1.00	0.09
IL13	1.00	1.00	1.00
IL17A	0.94	1.00	1.00
IL22	0.97	1.00	1.00
IL23A	0.59	1.00	0.76
TGFB1	0.00	0.01	0.00
FOXP3	0.00	0.00	0.00
IL10	0.00	0.08	0.00
IL8	0.00	0.10	0.01
LCN2	0.00	0.05	0.00
CAMP	0.00	0.00	0.00
DEFB4A	0.48	1.00	0.65
ELANE	0.07	1.00	0.12
EGF	0.00	0.01	0.00
CTGF	0.00	0.00	0.00
CSF2	0.54	1.00	0.71
IL9	0.76	1.00	0.89
COL1A1	0.00	0.00	0.00
COL1A2	0.00	0.00	0.00
COL4A1	0.00	0.00	0.00
COL4A2	0.04	1.00	0.07
ARG1	0.00	0.00	0.00
SPARC	0.00	0.00	0.00
MMP1	0.22	1.00	0.35
MMP2	0.00	0.00	0.00
MMP3	0.34	1.00	0.49
MMP7	0.73	1.00	0.87
MMP8	0.00	0.00	0.00
MMP9	0.00	0.00	0.00
MMP10	0.85	1.00	0.96
MMP11	0.95	1.00	1.00
TIMP1	0.00	0.09	0.00
TIMP2	0.00	0.07	0.00
TIMP3	0.48	1.00	0.65

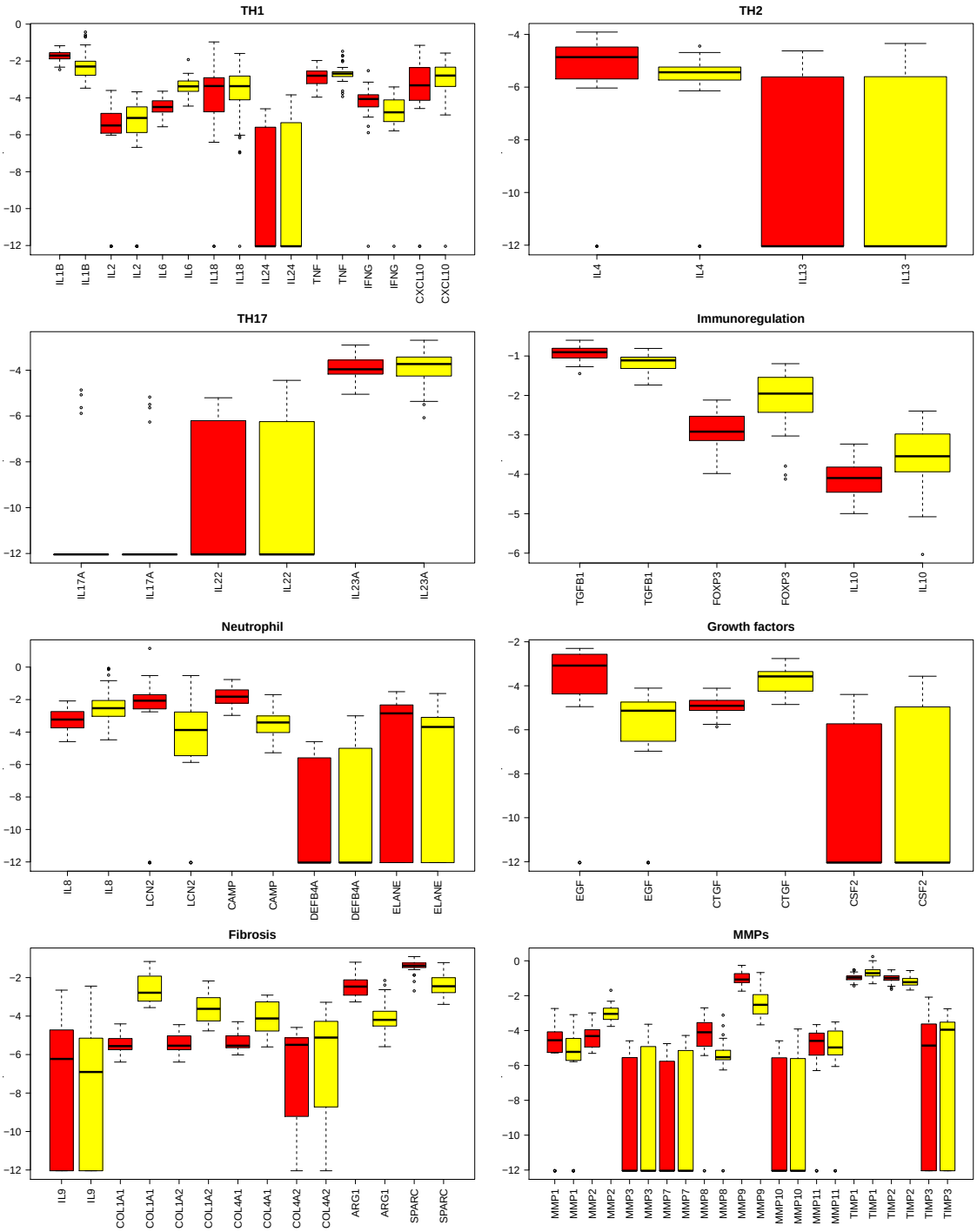


Figure 4.2: **Expression values by compartment.** Blood and pericardial fluid expression values for all 42 genes are shown, grouped into eight categories. Expression values are $\log_2$-fold change relative to $\beta - actin$, and as such all values are negative. More negative values correspond to lower expression.

and neutrophil-associated genes could be seen, as well as co-expression of some pro- and anti-inflammatory genes. The expression of several genes in fluid also exhibited negative correlation (e.g. IL-9 is negatively correlated with IL-2, IL-4, ELANE and MMP1)

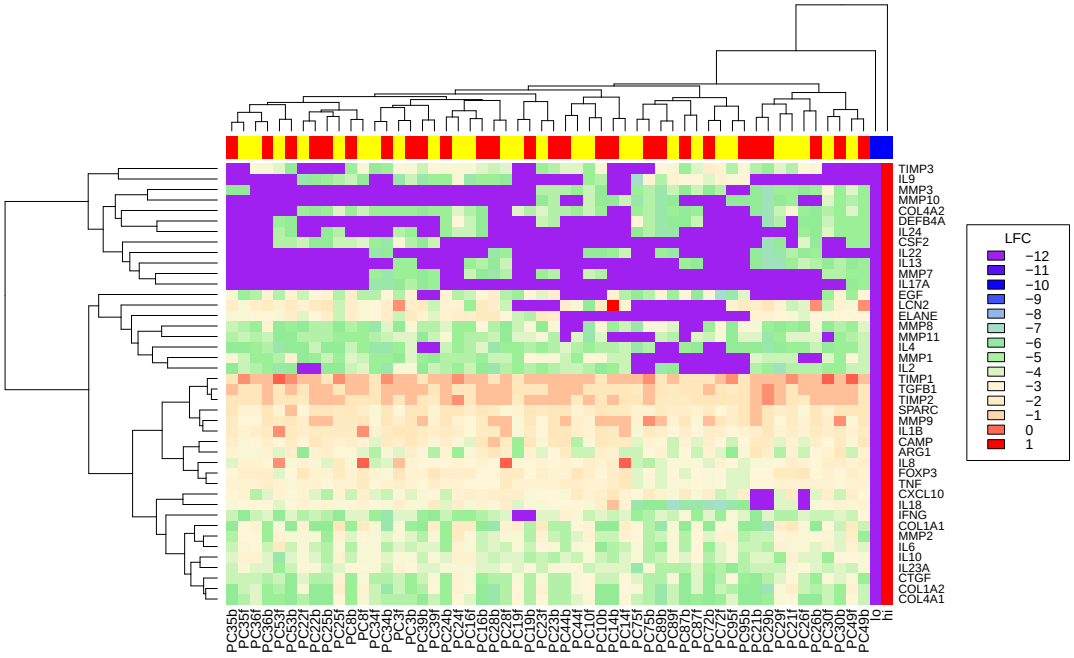
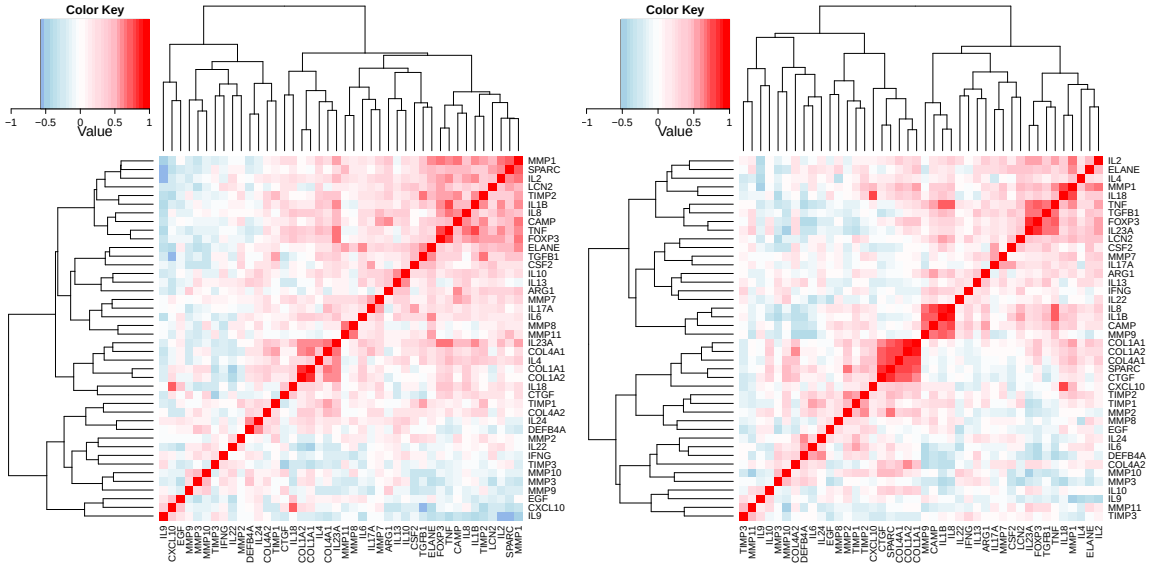


Figure 4.3: **Calibrated heatmap of all gene expression values.** This heatmap shows all expression values. Samples and genes are clustered by hierarchical clustering with average linkage. Strikingly, most blood and pericardial fluid samples from the same individual cluster together.

Given the very low expression levels of some genes, we next applied a non-specific filter to the data, which removed genes with delta-CT values greater than 38 in 5% or more of the samples (see Section 3.1.3). This step left 22 genes that separate blood and pericardial fluid to a large extent, with three samples misclassified (Figure 4.5). This result indicates that overall gene expression in the two compartments is very different and confirmed our previous suggestion that genes with low levels of expression drive the patient-specific clustering, shown earlier.

Following computation of differential gene expression with multiple testing correction using a linear models approach (see Section 3.1.3), 17 genes were found to be differentially expressed between the blood and pericardial fluid compartments. These genes are summarised in Table 4.3 and illustrated on a volcano plot in Figure 9.2. The calibrated heatmap shown in Figure 4.7 illustrates that the selected genes collectively cluster the samples into two separate compartments of



(a) **Correlation matrix for 42 genes (blood).** (b) **Correlation matrix for 42 genes (pericardial fluid).**

Figure 4.4: **Correlation matrices for 42 genes.** Colours indicate the value of R^2 for each pairwise correlation of gene expression (raw ΔCT values) in the range -0.5 to 1; *red* = positive correlation, *blue* = negative correlation.

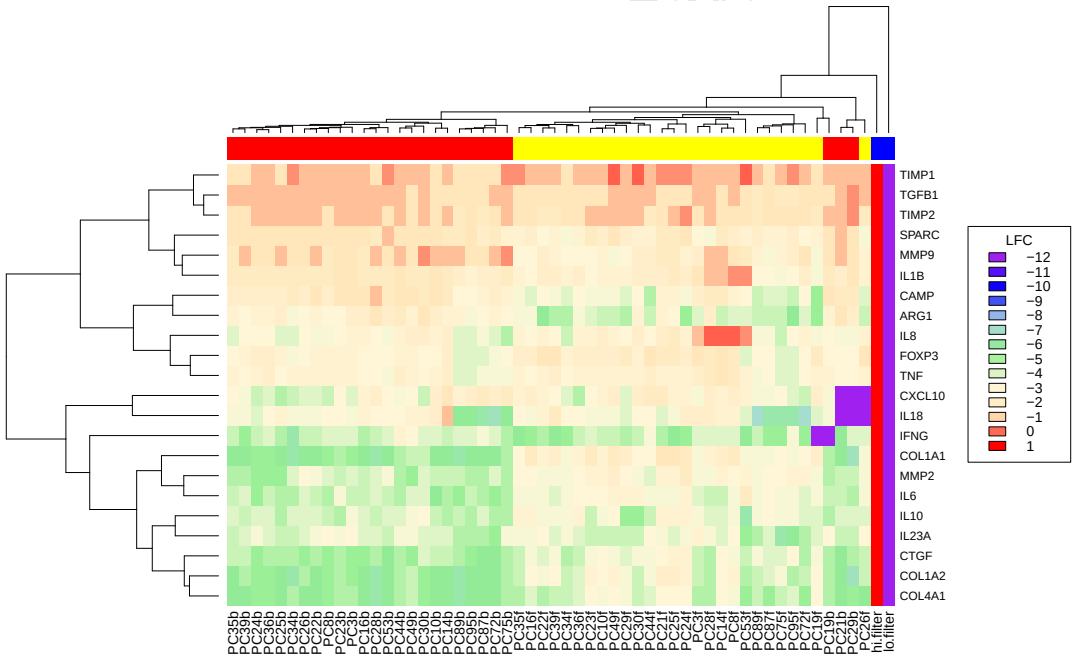


Figure 4.5: **Calibrated heatmap of all filtered expression values.** This heatmap shows expression values of filtered genes. Samples and genes are clustered by hierarchical clustering with average linkage. At this stage, blood and pericardial fluid form two separate clusters.

blood and pericardial fluid.

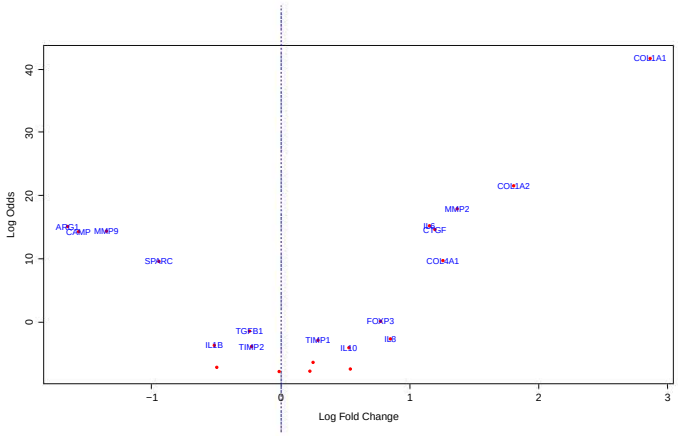


Figure 4.6: **Volcano plot of differentially expressed genes.**

Table 4.3: **Differentially expressed genes.** The logFC value represents increased or decreased transcript abundance in pericardial fluid relative to blood. A positive value indicates higher in fluid, and a negative value indicates higher in blood.

Differentially expressed genes			
ID	logFC	P.Value	adj.P.Val
COL1A1	2.86	6.51E-23	1.43E-21
COL1A2	1.80	3.41E-14	3.75E-13
MMP2	1.37	1.29E-12	9.46E-12
COL4A1	1.26	4.71E-09	1.15E-08
CTGF	1.19	3.40E-11	1.25E-10
IL6	1.15	1.89E-11	9.30E-11
IL8	0.85	1.49E-03	2.53E-03
FOXP3	0.77	8.07E-05	1.61E-04
IL10	0.53	6.67E-03	8.63E-03
TIMP1	0.29	1.87E-03	2.94E-03
TIMP2	-0.23	5.50E-03	7.56E-03
TGFB1	-0.25	4.08E-04	7.48E-04
IL1B	-0.52	4.44E-03	6.51E-03
SPARC	-0.95	5.32E-09	1.17E-08
MMP9	-1.36	4.38E-11	1.32E-10
CAMP	-1.57	4.80E-11	1.32E-10
ARG1	-1.66	2.11E-11	9.30E-11

Finally, we performed principal components analysis of the data using the three gene sets (all 42 genes, subset of 22 ‘filtered genes’ obtained after non-specific filtering, and the subset of 17 ‘selected genes’ obtained after differential gene expression analysis). The results are represented in Figure 4.8 (panels A, B, and C). While there was no separation of blood and fluid on the first two principal components using ‘all genes’ (panel A), the 22 ‘filtered genes’ (panel B) did indicate a

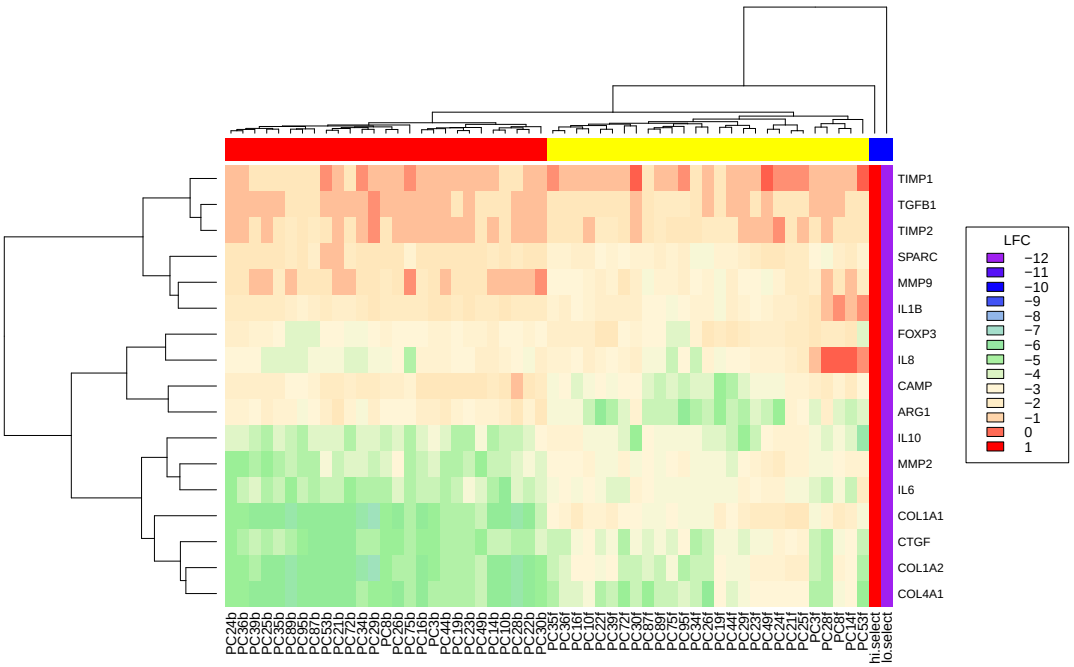


Figure 4.7: **Calibrated heatmap of differentially expressed genes.** This heatmap shows expression values of genes that are differentially expressed. Samples and genes are clustered by hierarchical clustering with average linkage. As expected, blood and pericardial fluid form two separate clusters.

separation of the two compartments. Finally, the first principal component of the 17 differentially expressed 'selected genes' (panel C) explained 60% of the observed variance.

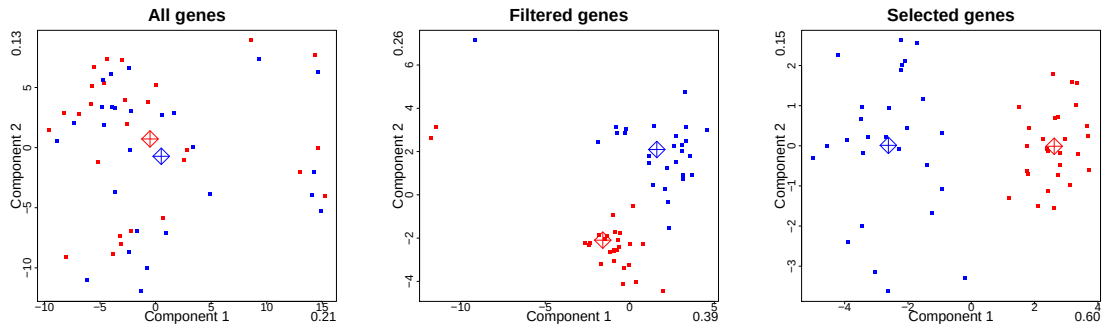


Figure 4.8: **Principal component analysis.** The first panel shows the results of principal component analysis of the whole dataset. There is no principal component that clearly allows for separation of blood and pericardial fluid samples. The second panel shows the same analysis applied to the filtered genes. Now, blood and fluid do separate, but not in a single dimension. Finally, PCA applied to only the differentially expressed genes shows clear separation of blood and pericardial fluid. Colours: *red* = blood; *blue* = pericardial fluid

4.4 Discussion

In this Chapter I demonstrate that significant differential abundance of gene transcripts can be detected in tuberculous pericarditis when contrasting blood and pericardial fluid. Transcripts associated with a Th1 response and fibrosis are more abundant in the pericardial fluid compartment, suggesting that larger-scale transcriptomic studies contrasting the blood transcriptome with that of the site of disease may be informative regarding pathogenetic mechanisms of tuberculosis.

Specifically, transcripts associated with fibrosis (COL1A1, COL1A2 and COL4A1) are more abundant, providing evidence at mRNA level of a profibrotic environment even without pericardial constriction present and suggesting that collagen synthesis takes place much in advance of clinical features of constrictive pericarditis. Increase of procollagen transcripts is accompanied by increase of transcripts of the growth factor CTGF. CTGF protein is secreted by vascular endothelial cells and promotes proliferation of chondrocytes, mediates cell adhesion in many cell types and enhances fibroblast growth factor-induced DNA synthesis.

ARG1 and SPARC, two negative regulators of fibrosis are less abundant at transcript level, again

underscoring the overall pro-fibrotic environment in pericardial fluid.

In addition to the pro-fibrotic environment in pericardial fluid there is some support for a Th1-mediated inflammatory response in the pericardial compartment, with increase of IL-6 transcripts in pericardial fluid. Other Th1-related markers are not increased, though, so this increase should be interpreted with caution. The increase of IL-10 and FOXP3 transcripts may indicate a concurrent anti-inflammatory response. Interpretation of pro-and anti-inflammatory responses in tuberculosis is not without difficulty, as it is unclear whether Th1 responses are protective, injurious, or both, and whether anti-inflammatory responses are protective or detrimental in the context of inflammation.

Neutrophil-associated genes IL-8 and CAMP are up-and downregulated in pericardial fluid, respectively; this does not provide a clear indication of the role of neutrophil-specific activity in tuberculous pericarditis.

Finally, matrix metallproteinases and their negative regulators appear to be dysregulated, with upregulation of MMP2 and TIMP1 in pericardial fluid, and concurrent downregulation of MMP9 and TIMP1.

This proof-of-concept analysis demonstrates that informative and biologically plausible results may be generated from analysis of matched blood and site-of-disease gene expression data.

5 Technical results

Chapter summary

In this chapter I provide an overview of all patients included in IMPI-MA, and develop the idea of ordering the study subjects along the vertices of multi-dimensional hypercubes in order to define contrasts and contexts for comparisons. This serves as the underpinning for a framework of analysing heterogeneous microarray data.

5.1 Included patients and samples

Figure 5.1 outlines the recruitment of study subjects into the IMPI-MA study, and the availability of RNA samples for these subjects. As stated previously, recruitment into the study consisted of three components: retrospective controls (EU Study), retrospective cases (recruited by Kerryn Matthews for her PhD work) and prospective cases (recruited by myself for this study). A total of 140 study subjects were recruited to contribute RNA samples for microarray analysis, and these individuals contributed a total of 220 blood and/ or pericardial fluid samples for quality assessment and microarrays. These samples were shipped to the NIMR in London. At the NIMR, the RNA was further processed and microarrays performed as described in Sections 3.6 and 3.7 . Following assessment of RNA yield and microarray quality metrics at the NIMR in London (see Figure 5.2), 39 samples were excluded, leaving 181 samples for analysis. Table 5.1 lists the reasons for exclusion of the 39 samples.

In order to plan the analysis, the 181 available microarrays were linked to phenotype information and stratified by three main criteria, based on the main hypotheses of the study: compartment (blood and pericardial fluid), HIV status (HIV+ and HIV-) and TB status (Healthy, LTBI, PTB and TB-PC).

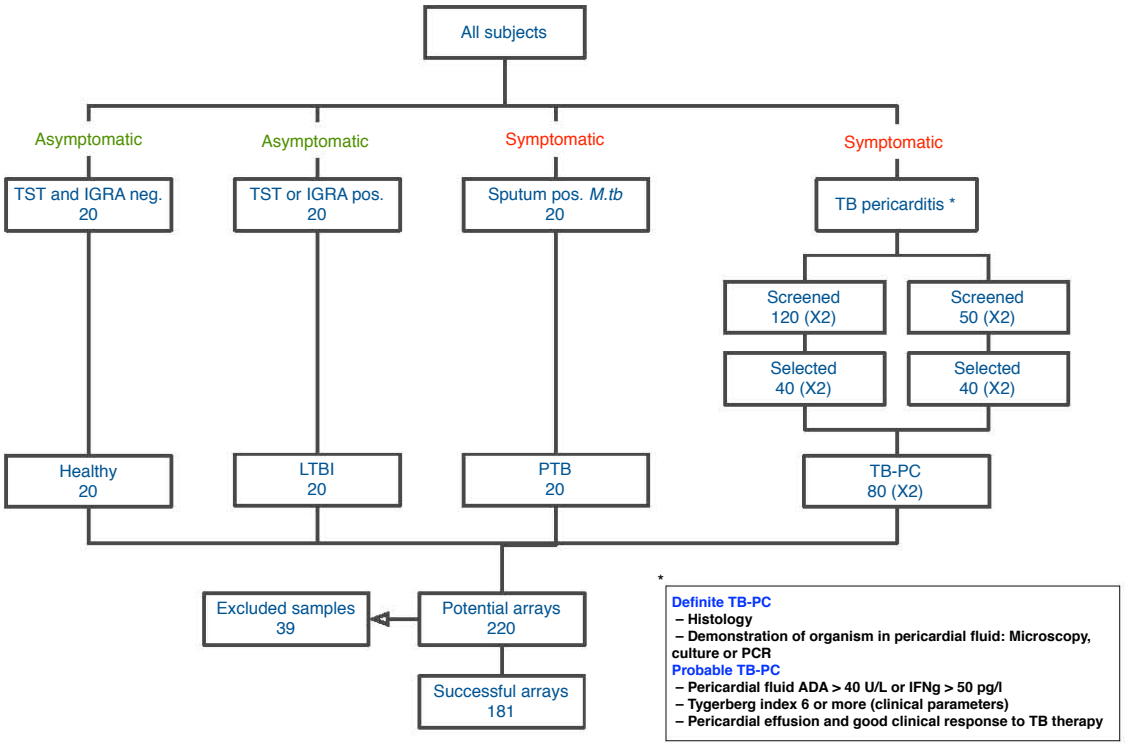


Figure 5.1: **Study subject recruitment.** The first three branches of the flowchart consist of the the subjects from the EU Study whose whole-blood derived RNA samples were included as controls (N=20 × 3. The TB-PC arm of the study consists of retrospective (N=40 matched blood and pericardial fluid) as well as prospective (N=40 matched blood and pericardial fluid) samples. The retrospective controls were selected on the basis of sample and data availability, the retrospective cases on the basis of sample and data availability as well as RNA quality (hence the large drop in numbers following screening), and the prospective samples on the basis of availability of matched blood and pericardial fluid samples. *Tygerberg index* refers to a score based on clinical (fever, weight loss, night sweats) and laboratory (serum globulin, blood leukocyte count) parameters [111].

Table 5.1: **Reasons for exclusion of RNA samples**

	Low yield (< 1250ng)	Low RIN (< 6)	Array QC fail	**Total**
TB-PC blood	7	3	0	**8**
TB-PC fluid	21	12	1	**24**
PTB	3	0	0	**3**
LTBI	2	0	0	**2**
"Healthy"	1	0	0	**1**
Total	**34**	**15**	**1**	**39**

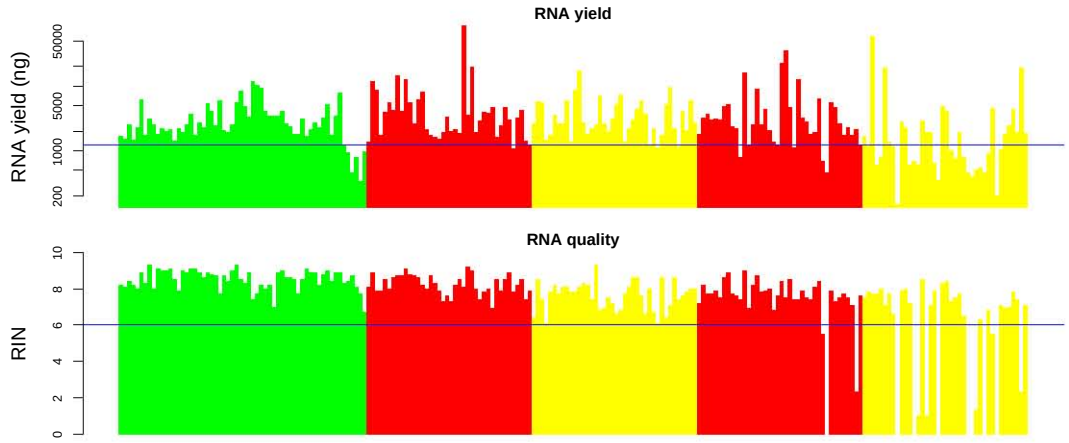


Figure 5.2: **RNA quality metrics.** The top panel shows the RNA yield (ng, in $\log_{10}$ scale) for each of the 220 samples, and the bottom panel the corresponding RNA quality (RIN, linear scale). Green bars correspond to EU study samples; red (blood) and yellow (pericardial fluid) bars correspond to retrospective (first pair) and prospective (second pair) TB-PC samples. Blue lines indicate pre-specified cutoff values used for including samples in further analysis.

The stratification of the 181 samples based on these criteria is shown in Figure 5.3.

5.2 Sample distribution along the vertices of a multi-dimensional hypercube

As is evident from Figure 5.3, the choice of possible contrasts and comparisons for analysis of this heterogenous and complex dataset is not immediately clear. The main problem lies with the fact that any *hierarchical* ordering of the sample subsets, as shown in the figure is completely arbitrary. Unsupervised analysis using principal component analysis may suggest an ordering, but even then the biological relevance of such a hierarchy is unclear. There is another option for representing the data, first discussed in Subsection 2.6.2, which allows for the fact that each phenotypic variable is independent of all others, and thus lies orthogonally oriented to them in phenotype space. Representing this graphically for three dimensions is trivial, as shown in Figure 2.3 for the three phenotype variables linked to the three main study hypotheses. Figure 5.4 (left panel) now shows this same 3-cube, while the right panel shows each of the cube's six faces, together with their respective sample numbers. The edges joining the four vertices of each face can now be interpreted as contrasts or comparisons (see Figure 2.2), where each contrast has a

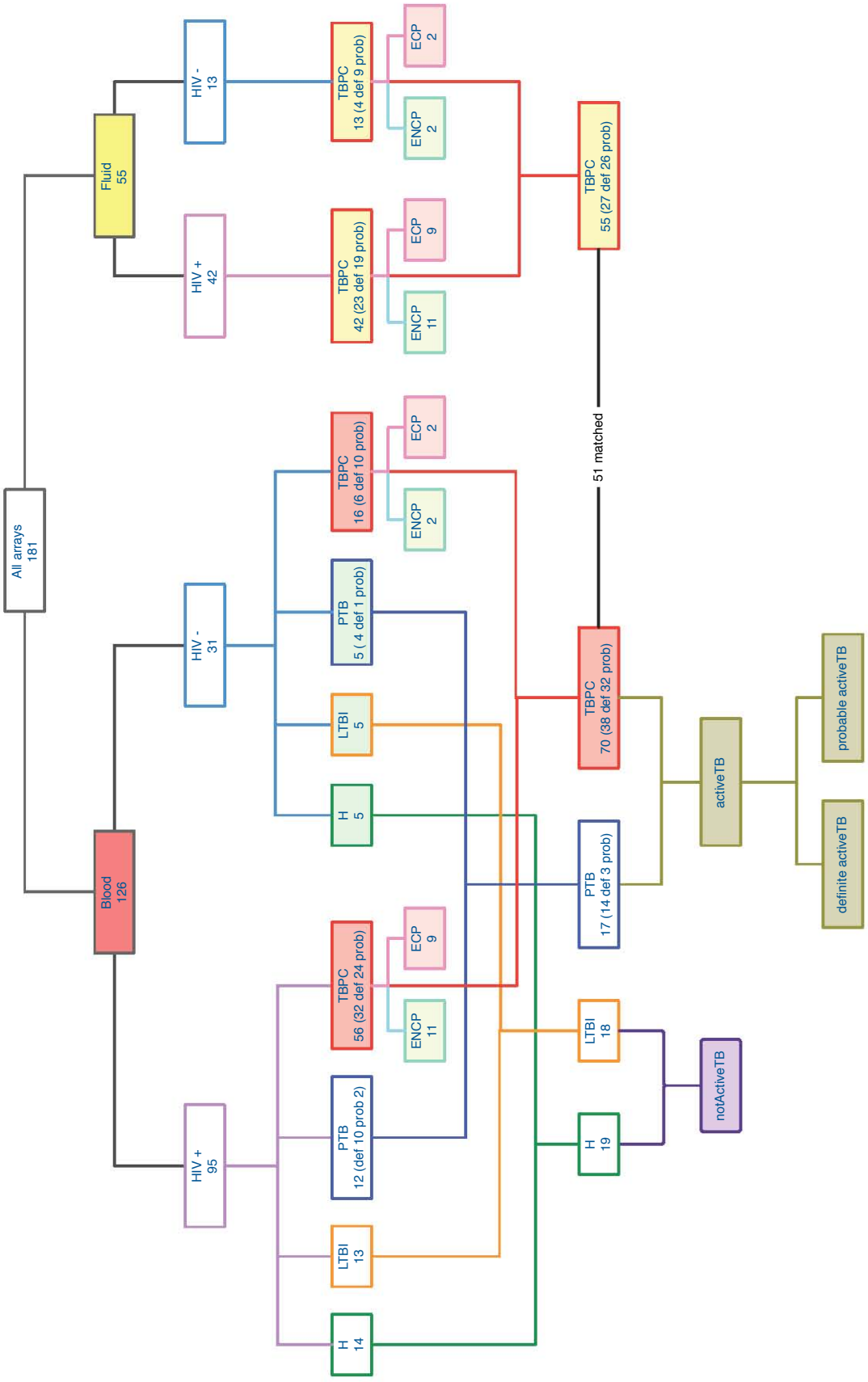


Figure 5.3: **Sample classes.** This figure illustrates the breakdown of sample numbers when stratified by three criteria (Compartment, HIV status and TB status). It further shows that subclasses may be recombined in different ways to yield additional groupings of potential interest for analysis.

Hamming distance [154] of 1, i.e. the phenotypes between the two vertices for each contrast differ by exactly one character, with all others equal.

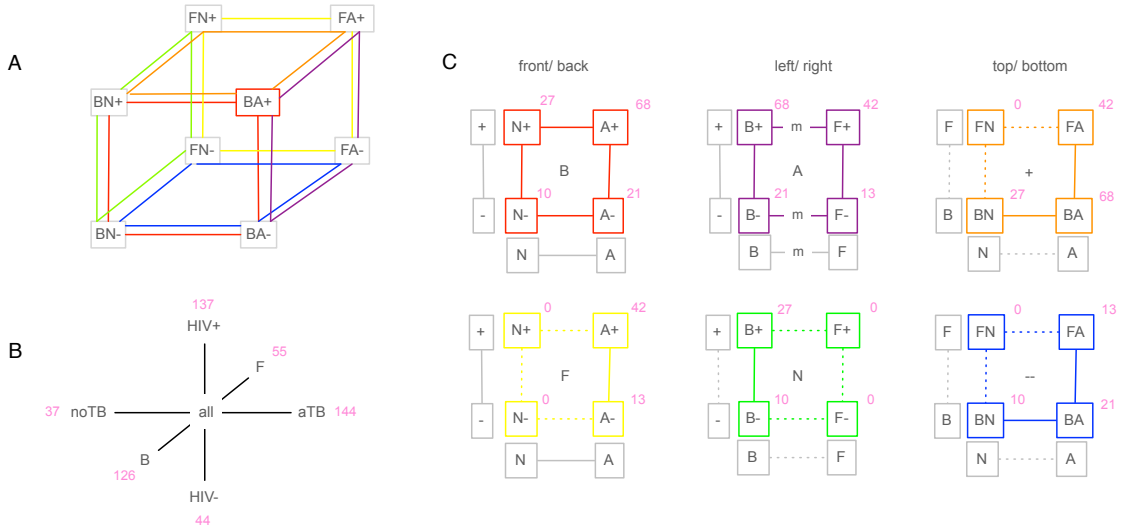


Figure 5.4: **Three-dimensional sample phenotype space.** The left panel shows the three-dimensional phenotype space relevant to the main study hypotheses (A) and the three orthogonal contrasts (B). The right panel (C) shows each face of the cube, with each vertex labeled by sample phenotype and number of available samples for that class. The edges represent potential contrasts or comparisons. Solid lines represent contrasts included in this study, and dotted lines represent contrasts excluded from the analysis due to insufficient numbers. Abbreviations: **B** blood; **F** pericardial fluid; **A** active tuberculosis; **N** not active tuberculosis; **+** HIV-1 infected; **-** HIV-1 uninfected; **m** matched samples exist

This idea can of course be extended into higher dimensions, where one additional dimension is required for each additional phenotype variable to be used as a contrast. As there are a number of interesting subgroups of phenotypes in the dataset that may shed additional light on HIV-TB, three additional dimensions were explored: effect of TB site (pulmonary vs pericardial TB), early TB (latent tuberculosis vs healthy) and haemodynamic phenotype in TB-PC (effusive non-constrictive vs effusive-constrictive). Figure 5.5 illustrates the resulting higher-dimensional 3-cubes, and Figure 5.6 illustrates how the additional dimensions may be mapped to a three-dimensional representation of a six-dimensional hypercube (hexeract). This structure contains 160 cubic cells, the four which are of interest are shown. To understand this more easily, ons should consider the original three-dimensional phenotype space, cube A in Figure 5.5. The left face of cube A consists of all samples mapping to the "not active TB" phenotype. This phenotype may be divided into

two groups (“Healthy” and “LTBI”) based on TST and IGRA results. This division “extends” the two-dimensional left face of cube A into a three-dimensional cube of its own (cube B, Figure 5.5). The six faces and associated sample numbers are shown in Figure 5.7. In contrast, the right face of cube A consists of all samples mapping to the “active TB” phenotype, which may be divided into two groups based on clinical presentation (“Pulmonary TB” and “Tuberculous pericarditis”), again “extending” the two-dimensional right face of cube A into cube C (Figure 5.5 and Figure 5.8). Finally, the right face of cube C (“Tuberculous pericarditis”) may yet again be “extended” into cube D based on haemodynamic phenotype (“effusive non-constrictive pericarditis” and “effusive-constrictive pericarditis”), shown in Figure 5.5 and Figure 5.9.

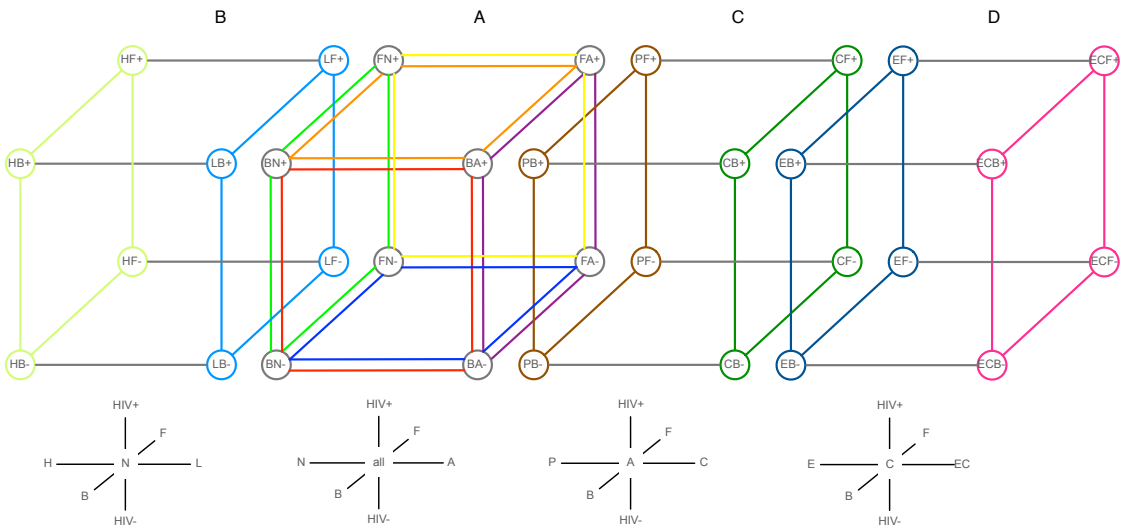


Figure 5.5: **Extension of three-dimensional cube to six dimensions.** Cube A is the cube originally discussed, and will serve as the reference point.

This concept of adding dimensions to a phenotype hypercube for each measured variable may be further generalised to very high dimensions. The full description of the “system state” of a complex system such as a human being requires a very large number of dimensions. This has important implications for sampling. When drawing a sample from the study population, one ideally requires a sample where samples for each contrast are present in equal numbers in each of the other contrasts, in order to avoid bias and confounding.

Such a sampling strategy would then result in a balanced dataset with (1) similar numbers of samples on all vertices of the n-dimensional hypercube, and (2) similar distributions for values of

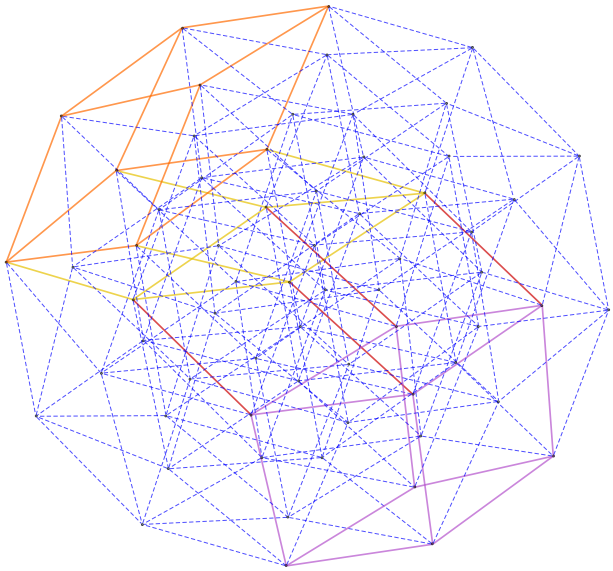


Figure 5.6: **Six dimensional hypercube.** The four cubes shown in Figure 5.5 may be mapped to vertices in six-dimensional phenotype space, shown here as a three-dimensional "shadow" of the six-dimensional space.

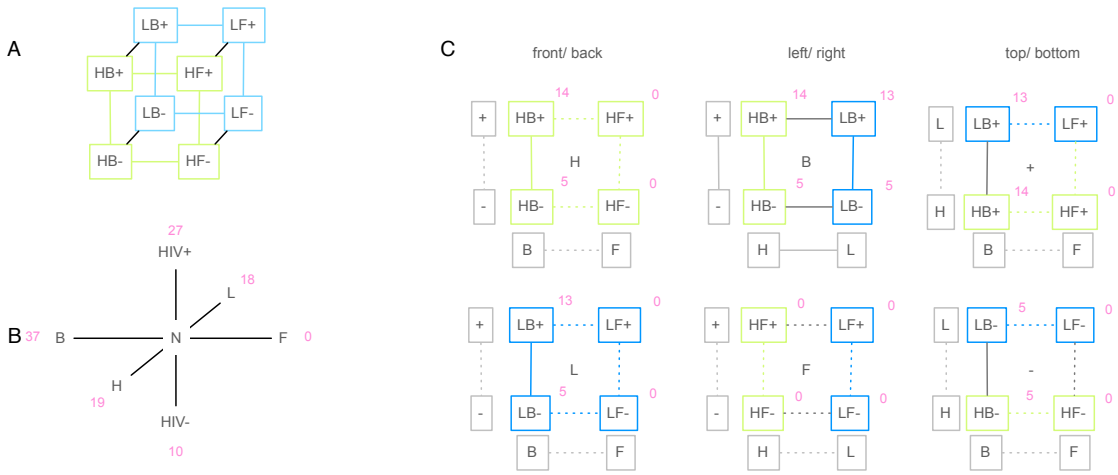


Figure 5.7: **Three-dimensional phenotype space for LTBI.**

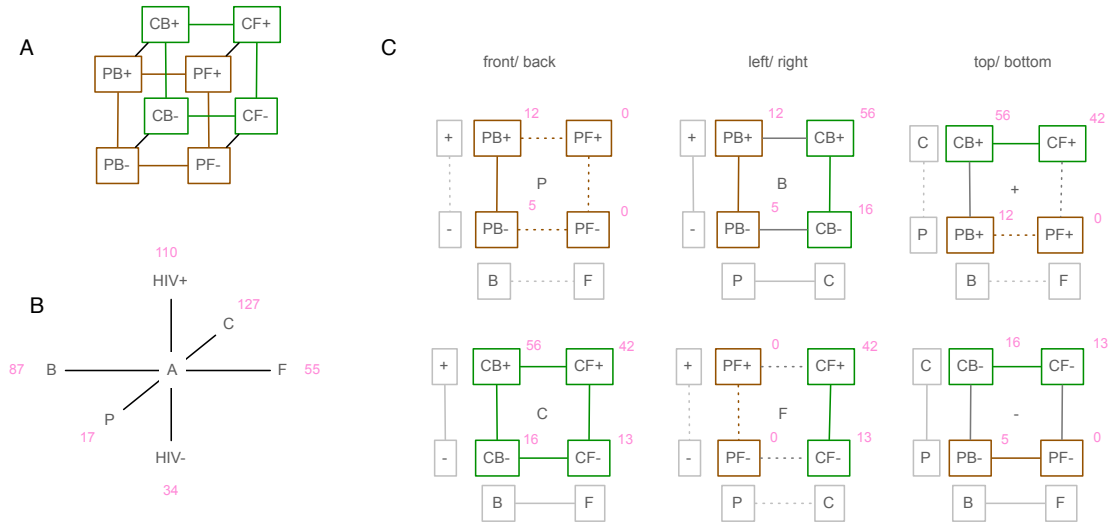


Figure 5.8: **Three-dimensional phenotype space for TB-site.**

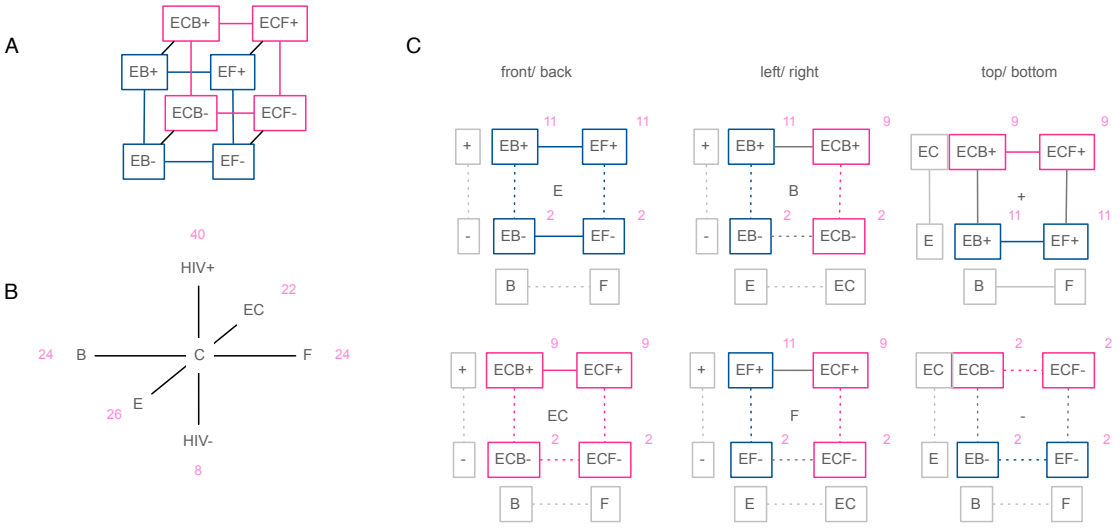


Figure 5.9: **Three-dimensional phenotype space for HD phenotype.**

all non-contrast variables (e.g. age, sex, ethnicity) for all vertices. By inspection we can verify from Figures 5.4, 5.7, 5.8 and 5.9 that the first criterion is not satisfied in three or higher dimensions, as the sample numbers for each vertex vary widely, and is often zero. Criterion (2), i.e. matching of non-contrast-variable distributions will be assessed for each contrast that was analysed by inspecting the table of clinical characteristics of each sample subset in the relevant results chapters of this thesis.

In summary, the visualisation of the data mapped to vertices of a multi-dimensional hypercube subgraph allows for data-driven selection of contrasts to examine. This concept will be re-visited in Chapter 8, where the analytic framework for the main study analysis will be presented.

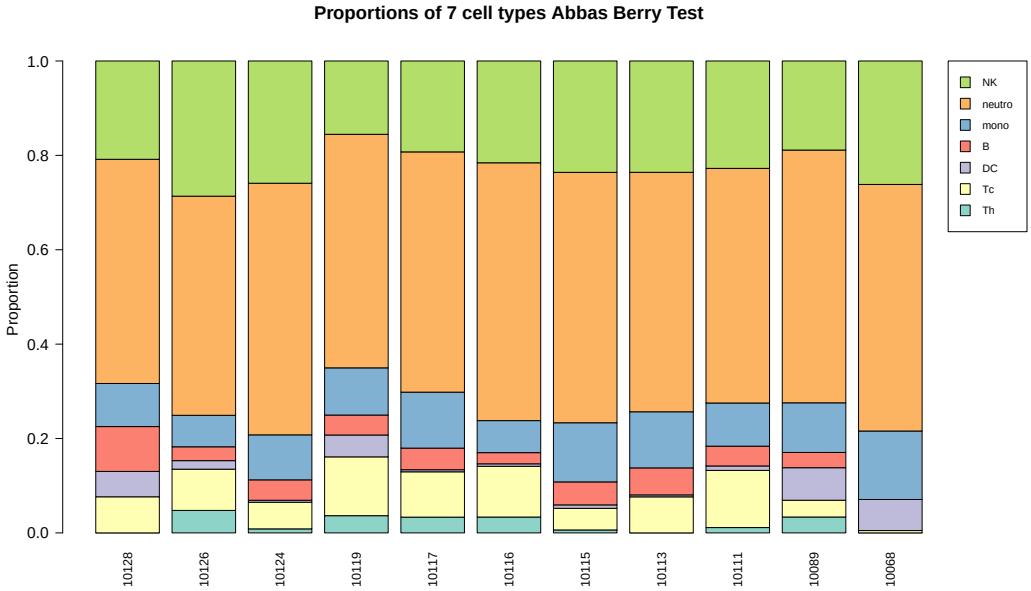
5.3 Validation of deconvolution approach: proof of concept data

As pointed out in section 3.8.6, an attempt at validating the deconvolution method using flow cytometry data was made. Figures 5.10 and 5.11 show the results.

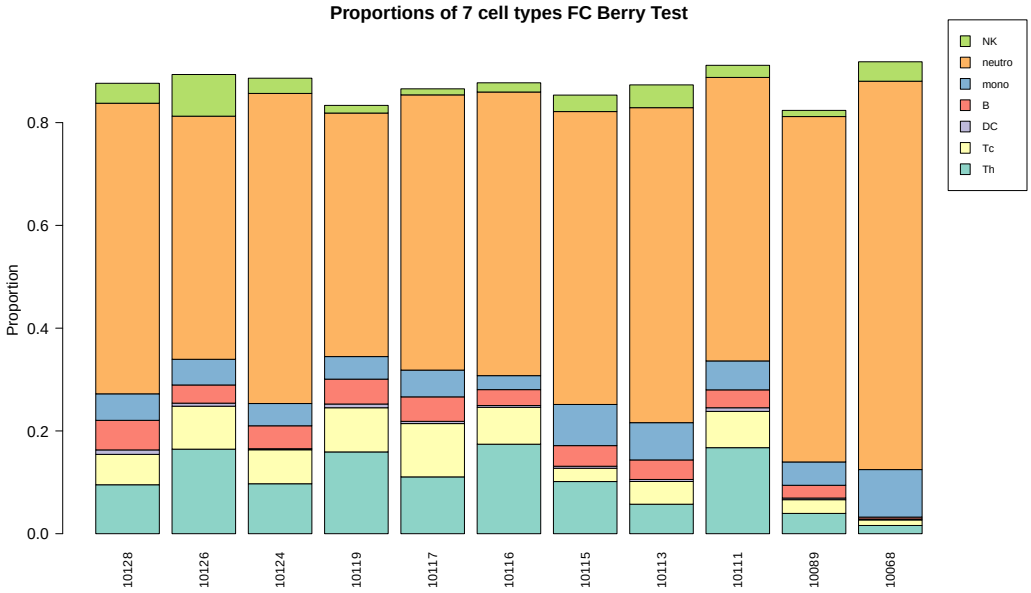
The barplots indicate an important feature of the data. All proportions obtained by deconvolution sum to one (by design) whereas the proportions obtained by flow cytometry do not.

Overall, the two methods correlate reasonably well (R^2=0.77). Some cell types (CD8 T cells, B cells, monocytes) correlate well, a second group (NK cells) less convincingly, and a third group (CD4 T cells, neutrophils and dendritic cells) correlates poorly. With the data available, the reasons for this are not clear. Dendritic cells and NK cells are observed at much higher proportions in the deconvolution data, whereas CD4 T cells and neutrophils are observed at higher proportions in the flow cytometry data.

The above results suggest systematic effects for these phenomena. A few potential reasons will be discussed below, but I should emphasise that this analysis should not be considered definitive. Firstly, neither of the proportion matrices contain "pure" proportions of single cell types but rather "composite" cell types obtained by adding proportions of different cell types together, as shown in Figure 3.10. Therefore the proportion signals do not necessarily represent the exact same cell populations. Secondly, in the deconvolution method used, the basis matrix based on the Abbas paper was generated on the Affymetrix platform, and mapping of transcript IDs (via Entrez IDs) to



(a) Proportions obtained by deconvolution



(b) Proportions obtained by flow cytometry

Figure 5.10: **Barplots of proportions in 11 samples** While the proportions in the top panel all sum to one (by design), the proportions in the bottom plot do not. Each sample therefore has an additional category of unassigned cells, which belong to one of the measured categories, or to a different category (e.g. basophils, eosinophils)

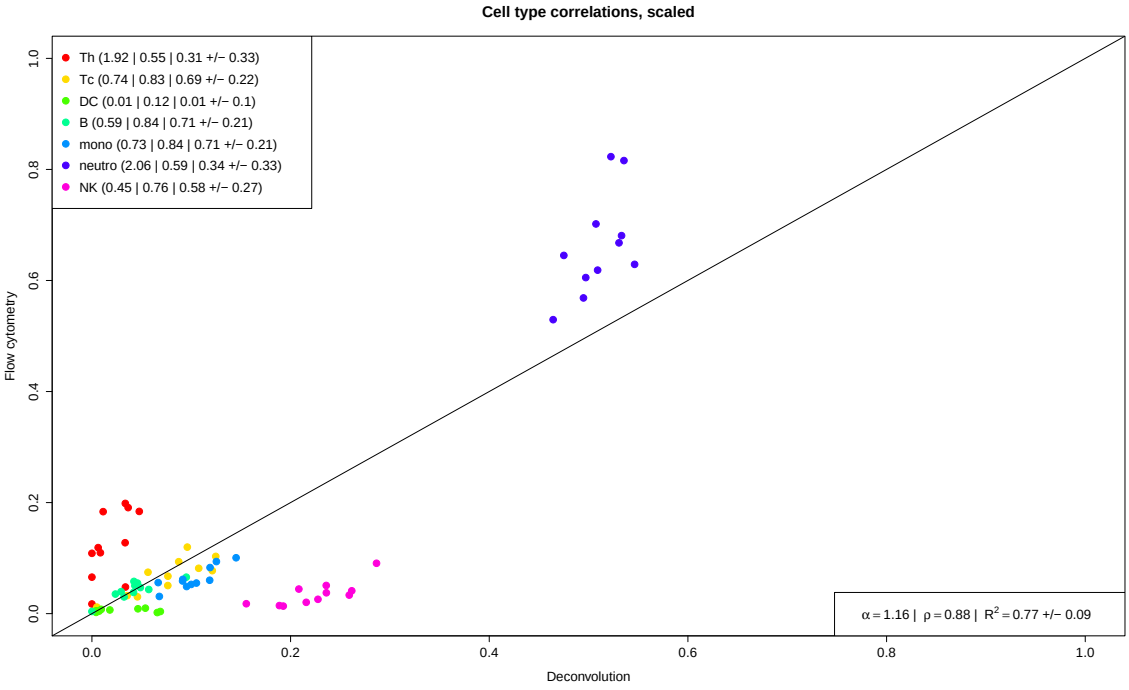


Figure 5.11: **Correlations of cell proportions** The *CellMix* function *profplot* was used to generate this plot. Prior to calculation of the correlations, the flow cytometry proportions matrix was rescaled to one, which mirrors the deconvolution matrix. Different cell populations are indicated by different colours. Clear systematic effects are visible for NK cells, dendritic cells, neutrophils and CD4 T cells. The values for the overall plot and the individual cell types represent the regression coefficient (α), Spearman rho (ρ) and Pearson R^2 with 95% confidence interval.

the Illumina data recovered only a part of the original basis matrix (324 features of a total of 515, i.e. 62.9%). This incomplete mapping may also influence the determination of cell proportions using deconvolution. Finally, $\log_2$-transforming and quantile normalising the expression data may also skew the reported proportions; indeed it has been reported that using non-transformed, non-normalised data achieves more accurate results (R. Gaujoux, personal communication).

In conclusion, the deconvolution method is promising, but requires further validation, ideally by generating a bespoke basis matrix based on locally generated flow cytometry data and single cell type expression profiles using Illumina HT12v4 arrays, followed by prospectively collecting samples from a diverse range of volunteers and subjecting these samples to whole-blood flow cytometry and Illumina microarray with deconvolution.

6 An unbiased view of all array data

Chapter summary

In this chapter I provide an overview of all IMPI-MA array data, for the purpose of defining overt patterns in the data and to identify potential batch effects caused by technical factors. I compare some aspects of this dataset to three datasets published previously[82], in order to argue that the IMPI-MA dataset is similar to this previously published data.

6.1 Data

6.1.1 IMPI-MA

All data was generated as described in Section 3.7. The data consists of summarised, background-corrected, non-normalised probe-level data for all 181 arrays. The term "IMPI-MA" will be used to refer to this data.

6.1.2 BERRY

The data consist of the original summarised probe-level data used for the analysis of the Berry et al paper [82], specifically the training, test and validation sets. In personal correspondence with the first author of that paper I was able to obtain the raw, non-normalised probe-level data; this was necessary as the publicly available version of the data includes some preprocessing steps in the form of average normalisation (see Section 3.9.4). The term "BERRY" will be used to refer to this data. Two BERRY subsets were included: the "training set" consisted of individuals with active tuberculosis, LTBI and individuals without evidence of prior immune sensitisation; the same groups were available in the IMPI-MA data. The BERRY validation set only included individuals

with active tuberculosis or LTBI, but the study participants were recruited in South Africa and therefore possibly more closely matched to the IMPI-MA data.

6.2 Quality control plots

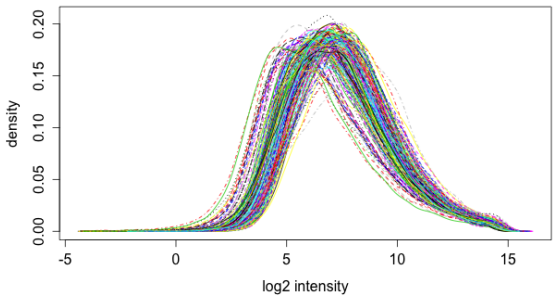
Figure 6.1 shows plots of the probability density function for six groups of arrays. This is a quick way to visually assess the similarity of expression profiles between arrays. As is clear from the plots, the overall shape of the probability density function (PDF) is the same for all arrays in each group but differs to some extent between groups of arrays. These differences are likely due to technical factors rather than biology, and indicate the need for a normalisation step. These plots highlight differences in array intensities for low to middle ranges of expression. In none of the plots are individual arrays visible that vary dramatically from the rest.

Figure 6.2 shows plots of the cumulative density function for six groups of arrays. Similarly to the PDF plots, these plots allow rapid assessment of all arrays for outliers in terms of overall array intensities. In contrast to the PDF plots, the CDF plots allow for better visualisation of intensity differences between arrays at the high and middle ranges of expression. Again, the need for normalisation becomes evident by inspection of the plots.

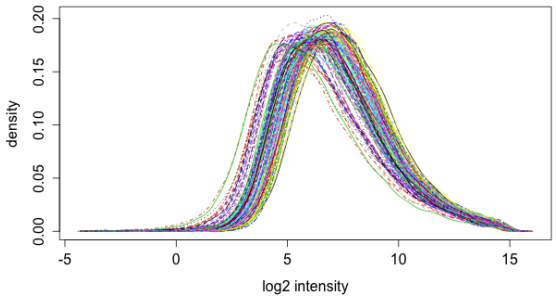
After reviewing the PDF and CDF plots we concluded that no samples needed to be excluded from analysis as no individual array differed dramatically from the rest. As the IMPI-MA and BERRY arrays looked significantly different, we concluded that direct comparison of these two studies might be problematic. Further comparison of the IMPI-MA and BERRY data is discussed in Chapter 7.

We now review some more aspects of the IMPI-MA data. In Figure 6.3 we show an example of pairwise correlation plots for five arrays from this dataset. These plots show that expression levels for individual probes for different arrays are highly correlated, an expected finding. A plot for all arrays was not done as the output would be illegible. As the data are not normalised, a high number of probes for each correlation pair differs by a factor of 2. Again, this expected.

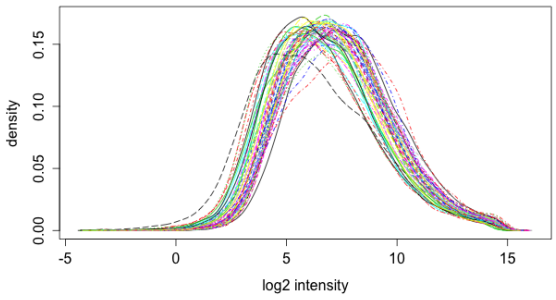
MA plots are used to visualise intensity-dependent ratios of raw microarray data; microarrays typically show a bias here, with higher A resulting in higher M, i.e. the brighter the spot the more likely an observed difference between sample and control. The MA plot uses M as the y-axis and



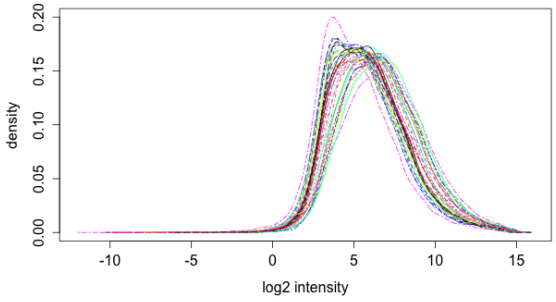
(a) **IMPI-MA: all arrays**



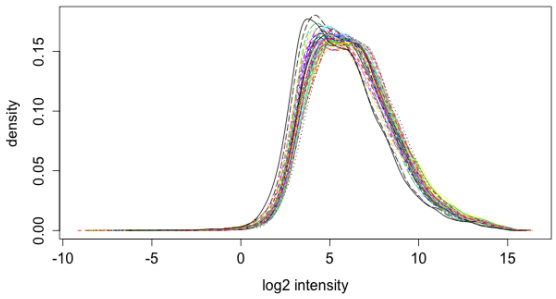
(b) **IMPI-MA: blood arrays**



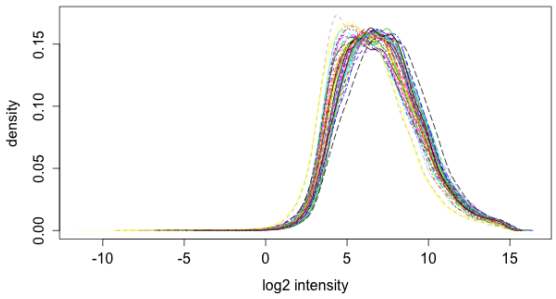
(c) **IMPI-MA: fluid**



(d) **BERRY: training set**

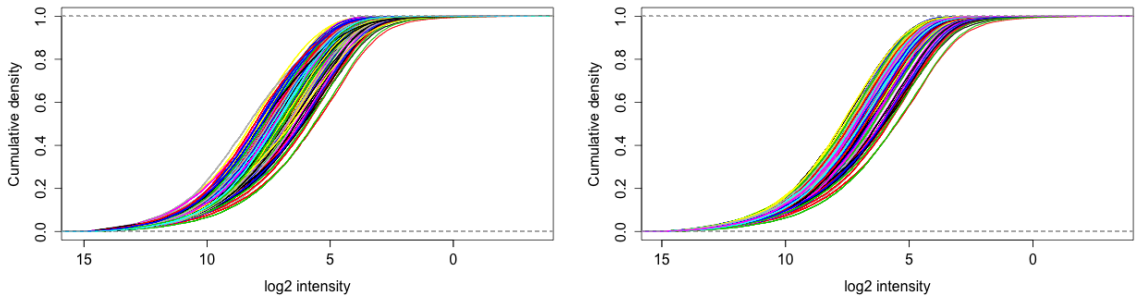


(e) **BERRY: test set**



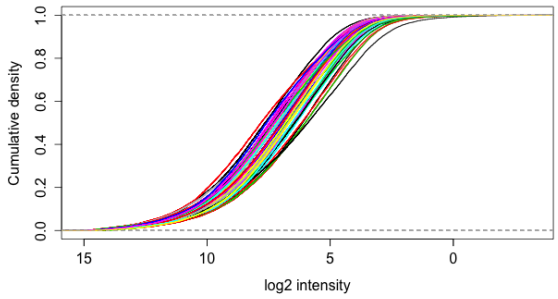
(f) **BERRY: validation set**

Figure 6.1: **PDF plots of all IMPI-MA arrays.** The first three plots show the superimposed probability density function (PDF) plots of all IMPI-MA data, and the blood and pericardial fluid subsets, separately. The fluid arrays in general appear to have similar expression distributions as blood, and important finding when considering batch effects. The last three plots show the PDFs for the BERRY training, test and validation sets, respectively. The x-axis shows the $\log_2$-intensity values, and the y-axis the corresponding probability for that intensity.

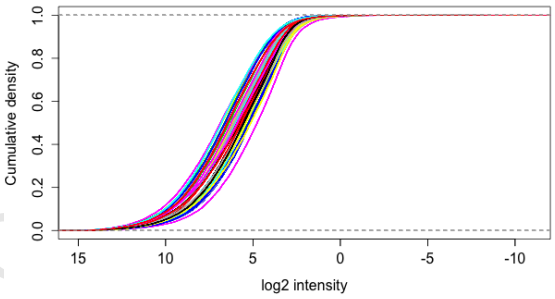


(a) **IMPI-MA: all arrays**

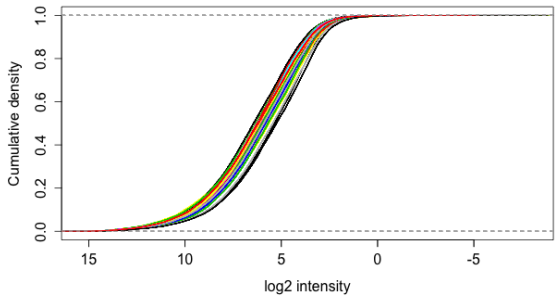
(b) **IMPI-MA: blood arrays**



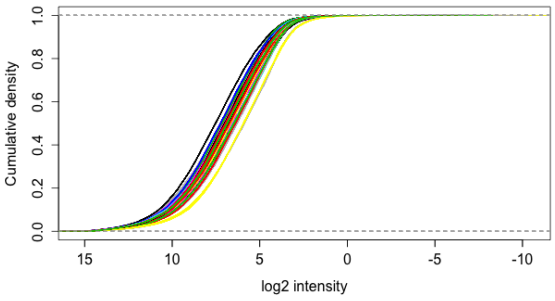
(c) **IMPI-MA: fluid arrays**



(d) **BERRY: training set**



(e) **BERRY: test set**



(f) **BERRY: validation set**

Figure 6.2: **CDF plots of all IMPI-MA arrays.** The first three plots show the superimposed cumulative density function (CDF) plots of all IMPI-MA data, and the blood and pericardial fluid subsets, separately. The last three plots show the CDF plots for the BERRY training, test and validation sets, respectively.

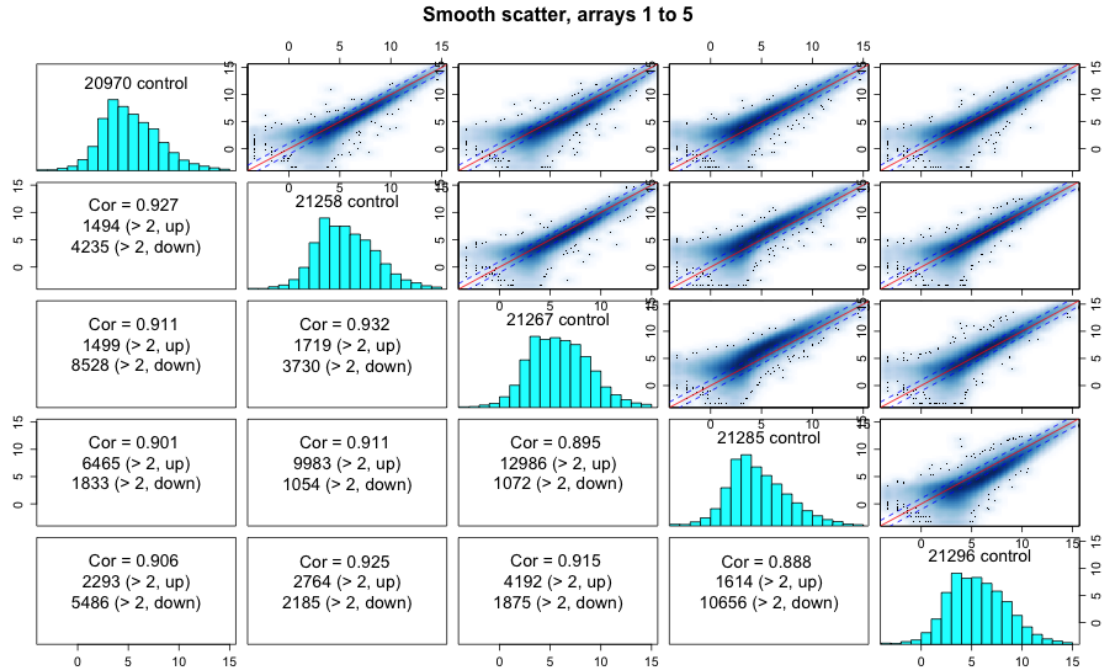


Figure 6.3: **Pairwise sample correlation.** Pairwise correlation of all probes on the first five arrays in the IMPI-MA dataset are shown.

A as the x-axis and gives another quick overview of the distribution of the data. Figure 6.4 shows M versus A (MA) plots for the first five arrays as an example. The M and A values are defined as follows:

$$M_k = \log_2\left(\frac{x_{ki}}{x_{kj}}\right) \tag{6.1}$$

$$A_k = \frac{1}{2}\log_2(x_{ki} \cdot x_{kj}) \tag{6.2}$$

The A values thus correspond to the mean of the two log values of two arrays i and j for all probes k on each array, while the M values correspond to the difference of the two log values. A Loess normalisation curve (shown in red) is fitted to the plot. In an ideal situation, this line should be straight. As it is not, it highlights the need for between-array normalisation.

Lastly, a box-and-whisker plot can highlight individual samples that differ significantly from all others. Such a plot is shown in Figure 6.5. The two ends of the box ("hinges") are versions

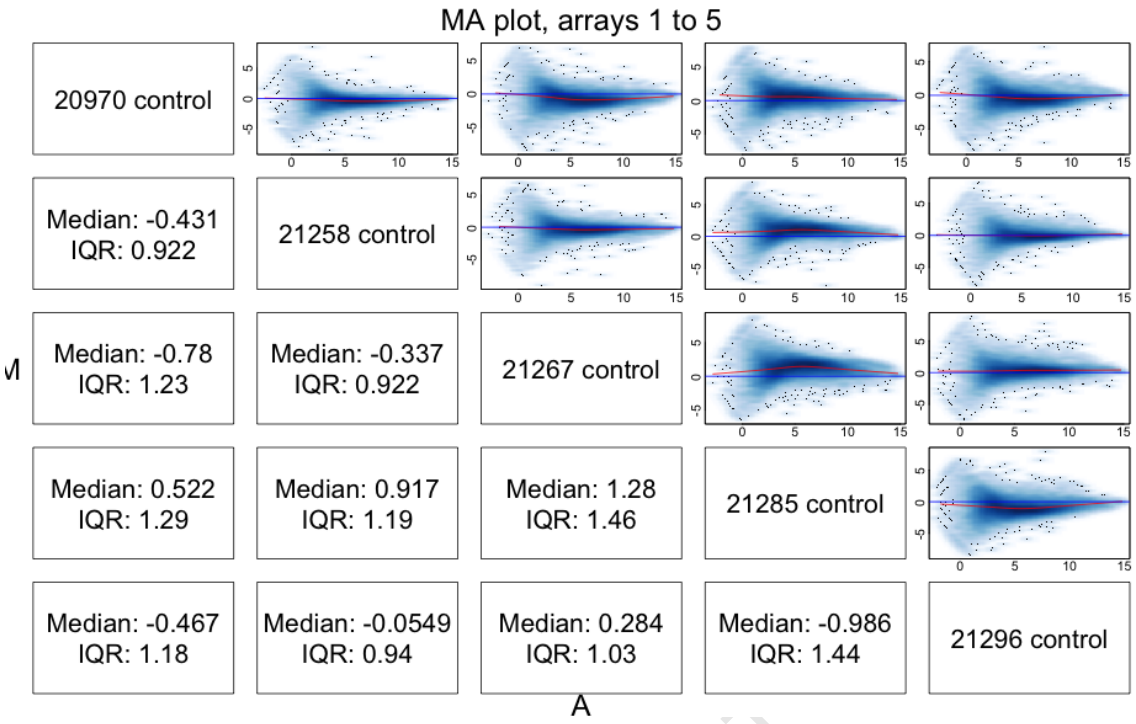


Figure 6.4: **MA plots for five arrays.** M versus A plots for the first five arrays in the IMPI-MA dataset are shown.

of the first and third quartile[1], and the "whiskers" extend to the most extreme data points. The plot shows that overall, the blood and pericardial fluid arrays have a (order of magnitude) similar median value, and that the median varies slightly between arrays, again indicating the need for array normalisation.

6.3 Sample relations

The previous subsection focused on various methods to assess overall intensity distributions in order to identify arrays that differed significantly from all others. This subsection now focuses on the discovery of systematic effects within the array data which may indicate that further downstream data analysis will be problematic. Relationships between the individual samples were assessed using a technique called metric multidimensional scaling (MDS). This methodology belongs to the class of methods called ordination, which aim to visualise information contained in

[1] The hinges equal the quartiles for odd n (where n <- length(x)) and differ for even n. Whereas the quartiles only equal observations for n mod 4 == 1 (n = 1 mod 4), the hinges do so additionally for n mod 4 == 2 (n = 2 mod 4), and are in the middle of two observations otherwise.

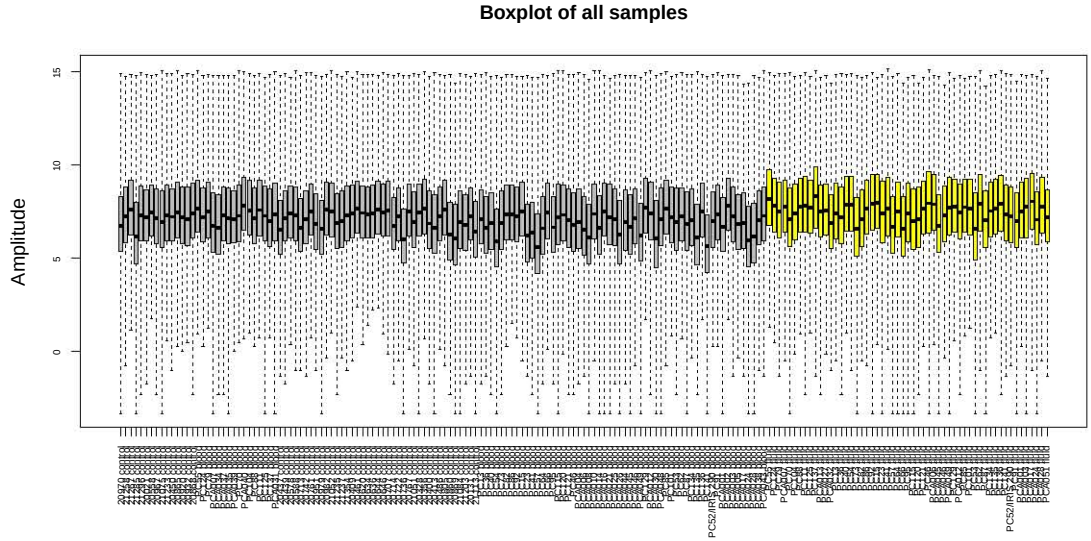


Figure 6.5: **Box-and-whisker plot of all IMPI-MA arrays.** Blood arrays are coloured grey, and pericardial fluid arrays are coloured yellow. Each box shows the upper "hinge", median and lower "hinge" for the $\log_2$-transformed expression data.

high-dimensional data matrices. Another well-known ordination technique is principal component analysis (PCA). MDS aims to place high-dimensional objects in n-dimensional space, such that the individual inter-object distances are preserved as accurately as possible. In this analysis we use two dimensions for the target space.

Microarray data are of very high dimension (approx. 47,000 in the case of summarised probe-level Illumina data), therefore dimension reduction techniques like MDS are useful for showing how close or distant two samples are to each other. While the algorithm for performing MDS will not be discussed here, some general principles are worth mentioning. The data object consist of the matrix of expression values for all probes and all samples. A distance matrix for all entries is computed (usually using a Euclidean metric), and this distance matrix is used to compute the two-dimensional representation of the data by the use of singular value decomposition. The result is a list of coordinates for each sample, on principal component axes. These coordinates preserve the distance information of the original data matrix, and can be plotted in a scatterplot. In this analysis, MDS is performed on all genes that are somewhat variable between samples (defined as

$^{standard\,deviation}/_{mean} > 0.1$).

PCA in contrast computes a covariance matrix of the input data and performs eigenvalue decomposition of that matrix, resulting in a list of principal components. The first principal component is defined as the one that accounts for as much variability in the data as possible (i.e. the normalised eigenvector with the largest associated eigenvalue of the sample covariance matrix). Each successive component is constrained by the requirement that it is orthogonal to the preceding one, and given this constraint explains as much of the remaining variance as possible. These two techniques are similar in several respects. While we only use MDS here to visualise data relationships, PCA will be used in later chapters of this thesis as well. In summary, MDS visualises dissimilarity between samples, and PCA identifies linearly uncorrelated components that explain most of the variance.

It is important to note that all of the analysis to follow was performed on non-normalised data. This was done to highlight potential technical bias. True biological differences between samples will be evaluated in later chapters, where that data has been normalised in order to deal with this technical bias.

Figure 6.6 shows the results of MDS analysis of all 181 microarrays. It is immediately clear that pericardial fluid arrays as a group behave very differently from blood arrays, and that the blood arrays do not show distinct subgroups based on TB status; indeed, there is complete overlap between the four categories (Healthy, LTBI, PTB and TB-PC).

MDS of blood samples is shown in Figure 6.7. Here, the samples are identified according to TB status and HIV Status. Again we see significant overlap of all groups, and no significant pattern suggesting another factor (e.g. technical) important in producing the observed differences.

Potential factors potentially biasing gene expression are sex [155] and the use of corticosteroids. Sex and corticosteroids may result in multiple gene expression changes that may bias differential expression analyses if the two groups contain unequal ratios of men to women. Figure 6.8 shows that no sex-specific or corticosteroid-specific signal appears to dominate the data.

In pericardial fluid, HIV status, and TB culture status (separating definite from probable TB-PC cases) may influence the overall pattern of gene expression. Figure 6.9 shows that neither TB culture status or HIV status dominate the overall expression data; differences that may exist will need to be sought by differential expression analysis.

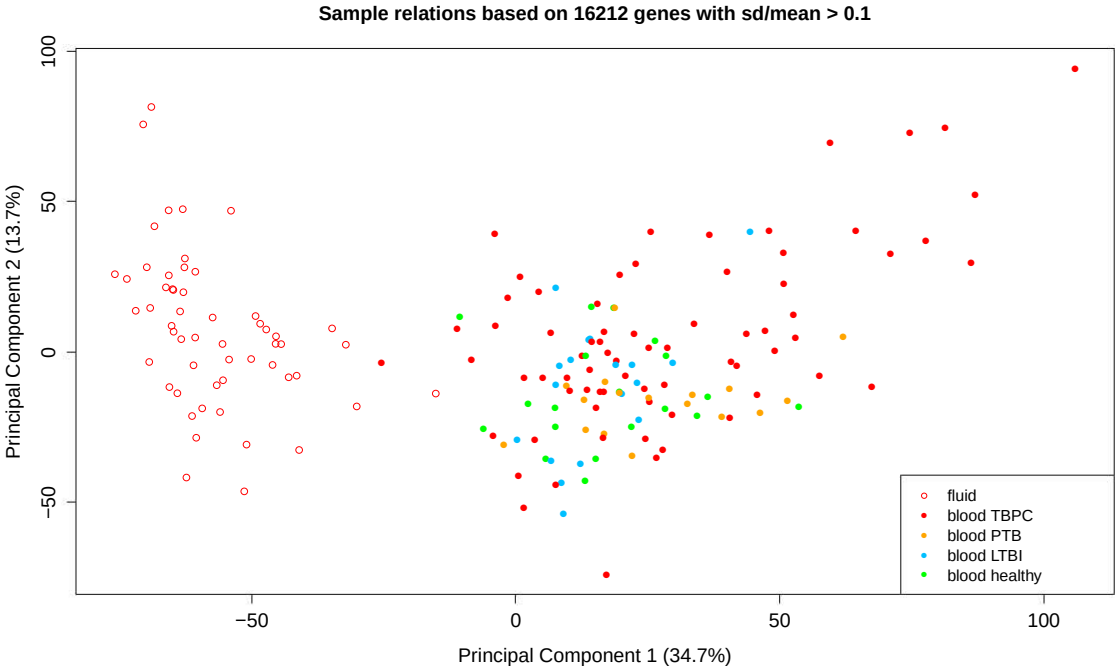


Figure 6.6: **MDS: All IMPI-MA data.** Each point consists of a sample. The sample classes are shown in the figure legend. There is striking dissimilarity between blood and pericardial fluid arrays.

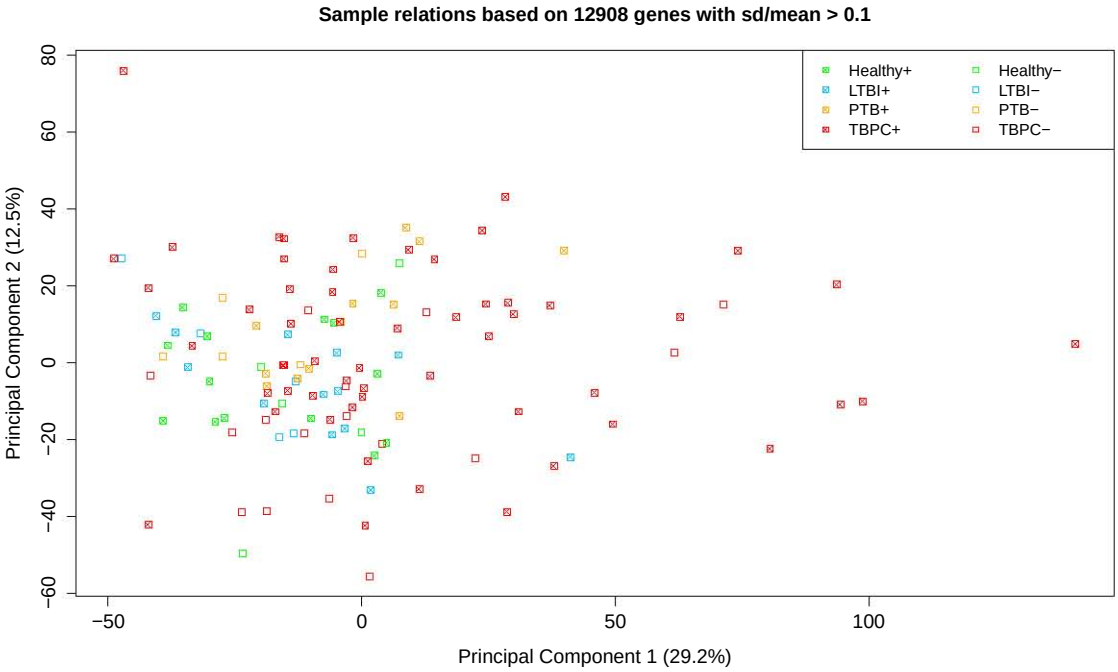


Figure 6.7: **MDS: IMPI-MA blood samples (HIV-TB).** Shown here are all blood samples, coloured by TB-status and HIV Status.

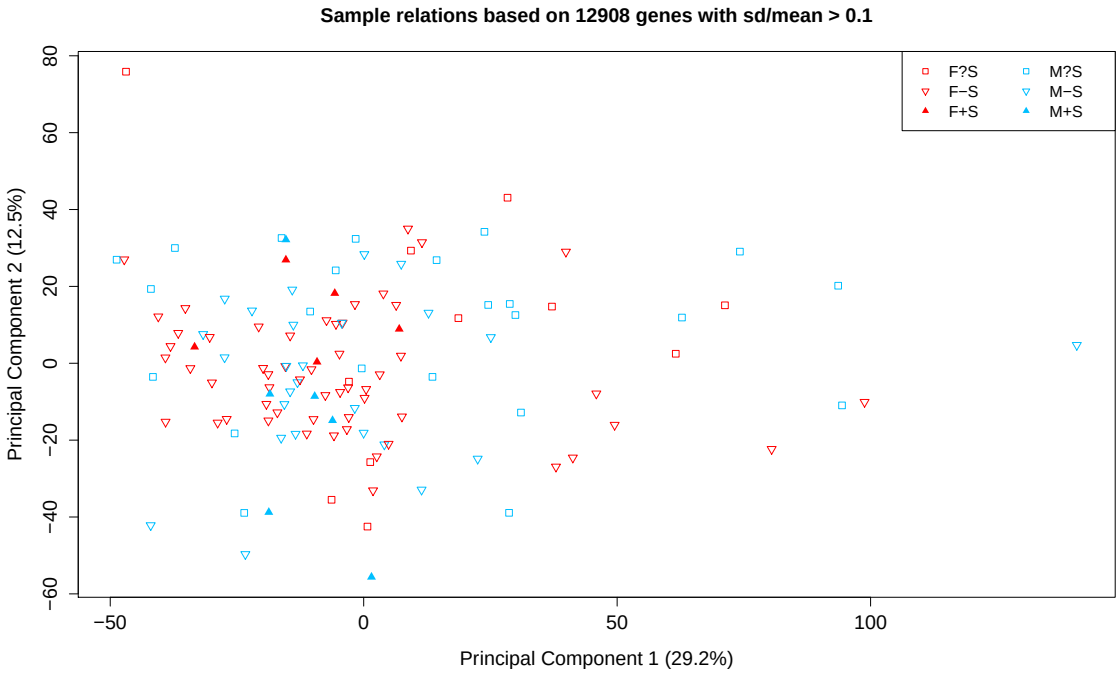


Figure 6.8: **MDS: IMPI-MA blood samples (Sex/steroids).** No clear separation appears between men (blue) and women (red), and between individuals who were taking corticosteroids (filled triangles) at the time of blood collection versus those who were not (hollow triangles). Squares represent samples where the steroid status was unclear.

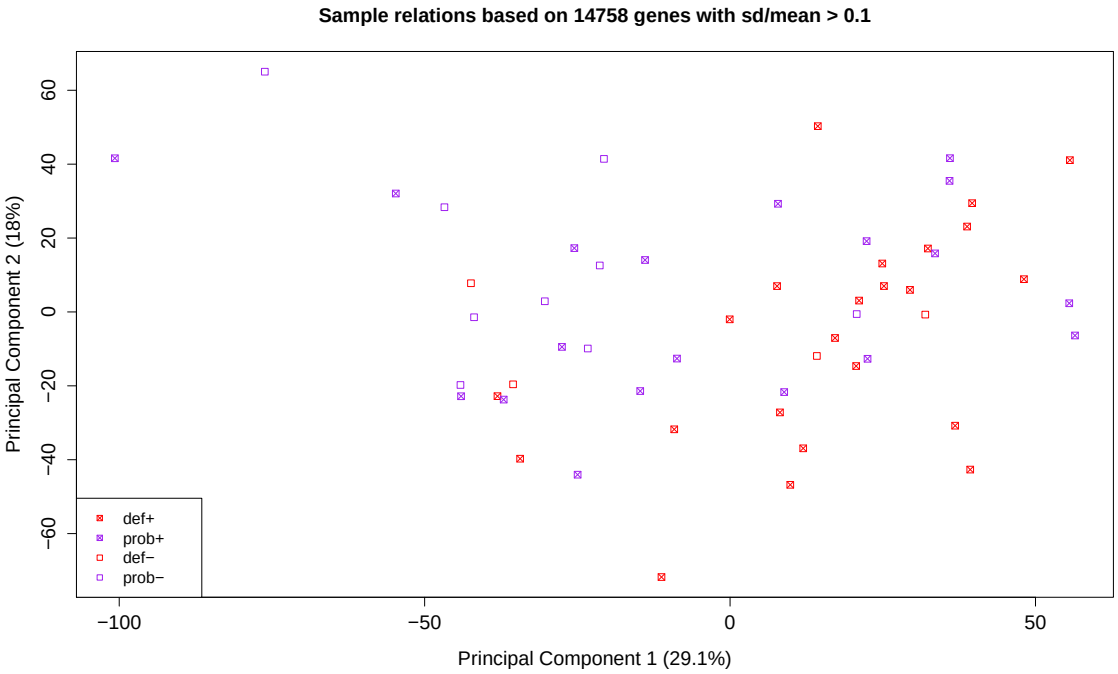


Figure 6.9: **MDS: IMPI-MA pericardial fluid samples (HIV-TB)** No clear separation appears between HIV positive (filled squares) and HIV negative (hollow squares) fluid samples, or between samples that were culture positive for *Mycobacterium tuberculosis* (red) versus those who were not (purple).

6.4 Discussion

The unbiased view of all IMPI-MA array data shows the data to be of high quality, and the need for normalisation between samples has been made evident. Multi-dimensional scaling views of the data clearly demonstrate that fluid samples as a whole behave differently from blood samples. Within the blood samples, I failed to detect gross technical bias. The individual sample classes do not separate on the two axes shown, despite the fact that the TB-PC and non-TB-PC samples were collected and processed at different times by different individuals. In addition, none of the parameters used to label the data showed any evidence of grouping by any of the studied parameters. This allowed me to proceed with confidence to subsequent analyses.

7 Relation of IMPI-MA data to previous studies of blood transcriptomics in tuberculosis

Chapter summary

In this chapter I demonstrate the results of an analysis pipeline in R that replicates the exact method used by Berry et al in the 2010 paper [82]. This is followed by application of the same method to a subset of the IMPI-MA data, and the results are compared.

7.1 Data

This analysis considers similarities in differential expression analyses of conceptually similar datasets. Therefore, the data included in this Section consists of the BERRY training and validation sets, and the subset of IMPI-MA data generated from blood samples in HIV-1 uninfected individuals who either had no clinical or microbiological evidence for active tuberculosis (Healthy, LTBI), or who had culture-confirmed tuberculosis (either PTB or TB-PC). The BERRY validation set is of interest, as the samples were collected in South Africa from a population demographically very similar to the IMPI-MA cohort. After due consideration, the BERRY training sets was selected in addition to the BERRY validation set as it was used to derive the 393 probe signature for active tuberculosis, and was closer to the IMPI-MA data in terms of the specified phenotype classes. The BERRY training set consisted of three sample classes (healthy, LTBI and PTB), the BERRY validation set only two (LTBI and PTB) and the IMPI-MA subset four (healthy, LTBI, PTB and TB-PC). Despite this, it is reasonable to expect that differential expression profiles contrasting active and

not-active tuberculosis should be comparable.

7.2 Study subjects

Table 7.1 shows comparisons between and within the three datasets across four clinical variables: age, sex, TST size and ethnicity. The comparison between datasets for all classes shows significant differences for age and ethnicity, but the datasets are matched for proportions of active to not active TB and sex. For each of the subsets (healthy, LTBI and active TB) we find the following: The *healthy* subset is mismatched w.r.t. ethnicity, the LTBI subset w.r.t. age, TST size and ethnicity, and the active TB subset w.r.t. ethnicity and TB type.

Comparisons within datasets find that the following datasets are not matched with respect to the following variables are not matched: BERRY training set: ethnicity; BERRY validation set: age. These differences between and within datasets are likely to influence differential expression analysis to some extent, but this is unavoidable.

7.3 Analysis

The raw data, following average normalisation looks similar across all three datasets (Figure 7.1).

When viewed using multidimensional scaling, none of the raw datasets show complete separation along the contrast of interest. Both the BERRY datasets show some extent of separation along at least one axis in the plot, suggesting a significant influence of the contrast variable.

Following the raw data visualisation, the data was processed as originally described in the supplementary methods section of [82]; see Section 3.9.4 for details of implementation and links to the relevant code. Table 7.2 shows the effects of applying the two non-specific filters to the three median-scaled data sets. Strikingly, 2.9 times the number of probes are selected from the IMPI-MA dataset (which used HT12 v4 arrays) when compared to the BERRY training set.

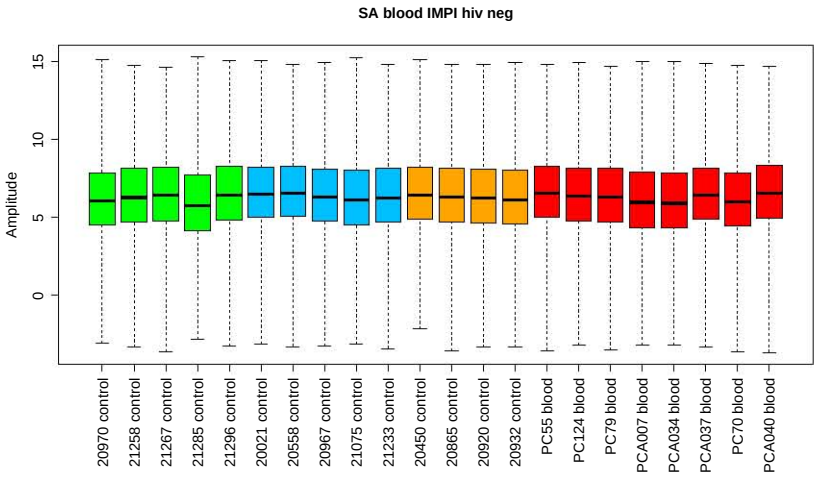
As the basis for the two filters may be dependent on technology, this may not be a fair comparison. For instance, newer technology may result in expression for more probes becoming detectable, and improved dynamic range of new array imaging technology may result in improved fold-change estimates. It is interesting that the BERRY validation set has results that are inter-

Table 7.1: **Table of clinical characteristics across datasets and contrasts** Abbreviations: **n**=number, **p**=proportion, **med**=median, **iqr**=interquartile range. P-values are given for comparisons *between datasets* (second-last column) as well as *within datasets* (last three rows), with the statistical test used for each comparison listed.

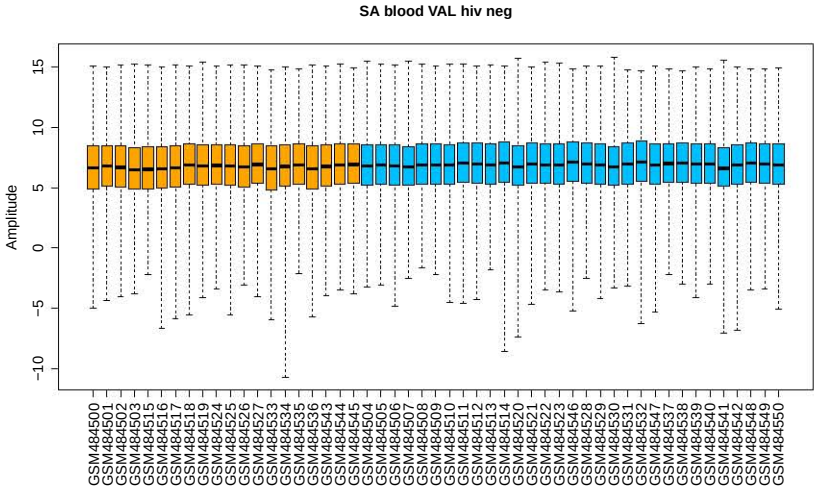
Subsets	variables	variable descriptors		BERRY training		BERRY validation		IMPI-MA		p value	test
All	N			42		51		22			
	age	med	iqr	30.5	25.0-35.0	24.0	21.0-27.0	33.9	26.55-51.48	1.341E-04	kruskal.test
	sex	n (f)	p (f)	23	0.55	25	0.49	7	0.32	2.124E-01	prop.test
		n (m)	p (m)	19	0.45	26	0.51	15	0.68		
	ethnicity	n (black)	p (black)	11	0.26	51	1.00	21	0.95	4.998E-05	fisher.test
		n (white)	p (white)	17	0.40	0	0.00	0	0.00		
		n (asian other)	p (asian other)	7	0.17	0	0.00	0	0.00		
		n (south asian)	p (south asian)	6	0.14	0	0.00	0	0.00		
		n (coloured)	p (coloured)	0	0.00	0	0.00	0	0.00		
		n (indian)	p (indian)	0	0.00	0	0.00	1	0.05		
		n (other)	p (other)	1	0.02	0	0.00	0	0.00		
	classes	n (active TB)	p (active TB)	13	0.31	20	0.39	12	0.55	1.850E-01	prop.test
		n (not active TB	p (not active TB)	29	0.69	31	0.61	10	0.45		
		n (healthy)	p (healthy)	12	0.29	0	0.00	5	0.23	4.998E-04	fisher.test
		n (LTBI)	p (LTBI)	17	0.40	31	0.61	5	0.23		
		n (PTB)	p (PTB)	13	0.31	20	0.39	4	0.18		
		n (TBPC)	p (TBPC)	0	0.00	0	0.00	8	0.36		
Healthy	N			12		0		5			
	age	med	iqr	31.0	27.75-34.25	NA	NA	30.7	23.8-32.7	7.503E-01	kruskal.test
	TST size	med	iqr	0.0	0.000-2.038	NA	NA	0.0	0-0	2.346E-01	kruskal.test
	sex	n (f)	p (f)	8	0.67	NA	NA	1	0.20	2.212E-01	prop.test
		n (m)	p (m)	4	0.33	NA	NA	4	0.80		
	ethnicity	n (black)	p (black)	0	0.00	NA	NA	5	1.00	4.998E-04	fisher.test
		n (white)	p (white)	12	1.00	NA	NA	0	0.00		
		n (asian other)	p (asian other)	0	0.00	NA	NA	0	0.00		
		n (south asian)	p (south asian)	0	0.00	NA	NA	0	0.00		
		n (coloured)	p (coloured)	0	0.00	NA	NA	0	0.00		
		n (indian)	p (indian)	0	0.00	NA	NA	0	0.00		
		n (other)	p (other)	0	0.00	NA	NA	0	0.00		
LTBI	N			17		31		5			
	age	med	iqr	32.0	25.0-39.0	21.0	19.5-24.0	29.1	22.7-34.7	7.253E-05	kruskal.test
	TST size	med	iqr	20.0	18.0-27.0	14.0	6.0-15.0	12.0	8.0-15.0	7.933E-05	kruskal.test
	sex	n (f)	p (f)	9	0.53	20	0.65	1	0.20	1.645E-01	prop.test
		n (m)	p (m)	8	0.47	11	0.35	4	0.80		
	ethnicity	n (black)	p (black)	7	0.41	31	1.00	5	1.00	4.998E-04	fisher.test
		n (white)	p (white)	2	0.12	0	0.00	0	0.00		
		n (asian other)	p (asian other)	5	0.29	0	0.00	0	0.00		
		n (south asian)	p (south asian)	3	0.18	0	0.00	0	0.00		
		n (coloured)	p (coloured)	0	0.00	0	0.00	0	0.00		
		n (indian)	p (indian)	0	0.00	0	0.00	0	0.00		
		n (other)	p (other)	0	0.00	0	0.00	0	0.00		
TB	N			13		20		12			
	age	med	iqr	29	23.0-33.0	31.5	25.0-42.0	40.67	33.69-52.86	8.391E-02	kruskal.test
	sex	n (f)	p (f)	6	0.46	5	0.25	5	0.42	4.055E-01	prop.test
		n (m)	p (m)	7	0.54	15	0.75	7	0.58		
	ethnicity	n (black)	p (black)	4	0.31	20	1.00	11	0.92	9.995E-04	fisher.test
		n (white)	p (white)	3	0.23	0	0.00	0	0.00		
		n (asian other)	p (asian other)	2	0.15	0	0.00	0	0.00		
		n (south asian)	p (south asian)	3	0.23	0	0.00	0	0.00		
		n (coloured)	p (coloured)	0	0.00	0	0.00	0	0.00		
		n (indian)	p (indian)	0	0.00	0	0.00	1	0.08		
		n (other)	p (other)	1	0.08	0	0.00	0	0.00		
	TB type	n (PTB)	p (PTB)	13	1.00	20	1.00	4	0.33	1.548E-06	prop.test
		n (TB-PC)	p (TB-PC)	0	0.00	0	0.00	8	0.67		
Technical	N			42		51		22			
	RNA type	n (Tempus)	p (Tempus)	42	1.00	51	1.00	0	0.00	2.200E-16	prop.test
		n (paxGene)	p (paxGene)	0	0.00	0	0.00	22	1.00		
	Array type	n (HT12v3)	p (HT12v3)	42	1.00	51	1.00	0	0.00	2.200E-16	prop.test
		n (HT12v4)	p (HT12v4)	0	0.00	0	0.00	22	1.00		
	age_within	p value	test	0.6616	kruskal.test	1.35E-06	kruskal.test	0.2276	kruskal.test		
	sex_within	p value	test	0.5775	prop.test	0.01354	prop.test	0.5542	prop.test		
	ethnicity_within	p value	test	0.0004998	fisher.test	NA	prop.test	1	fisher.test		

Table 7.2: **Effect of non-specific filtering.** Striking differences in the numbers of probes passing the filters can be observed.

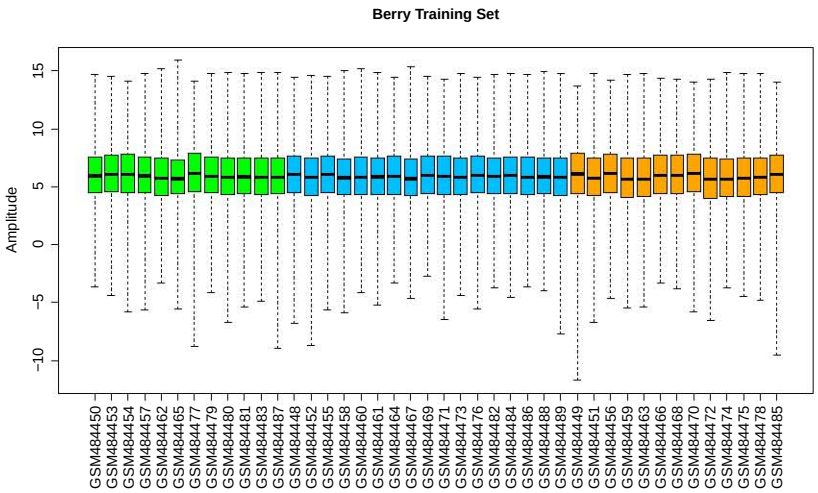
Dataset	Original number	Pass detection filter	Pass Fold change filter	Pass both filters
IMPI-MA	47,231	19,027	5,316	5,316
BERRY validation	48,802	15,418	3,388	2,824
BERRY training	48,802	15,391	1,836	1,835



(a) **IMPI-MA**

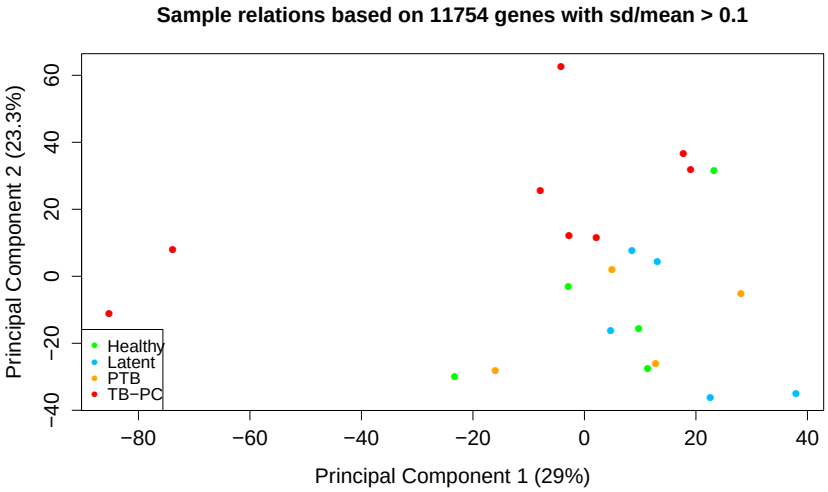


(b) **BERRY validation**

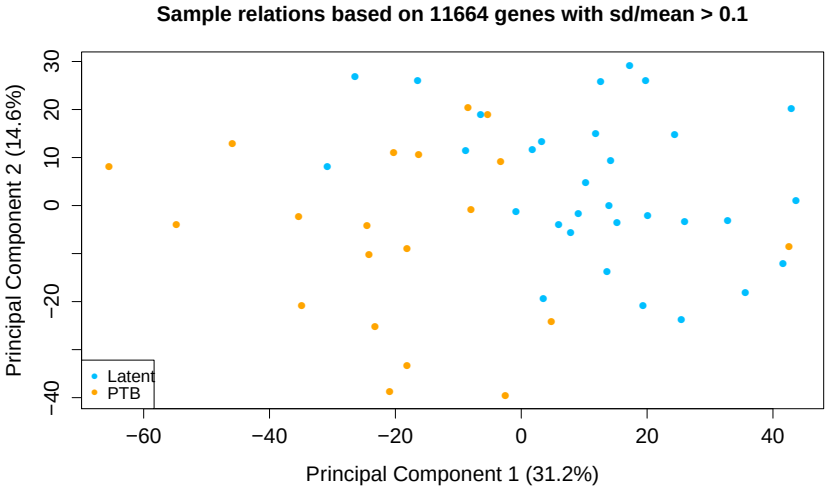


(c) **BERRY training**

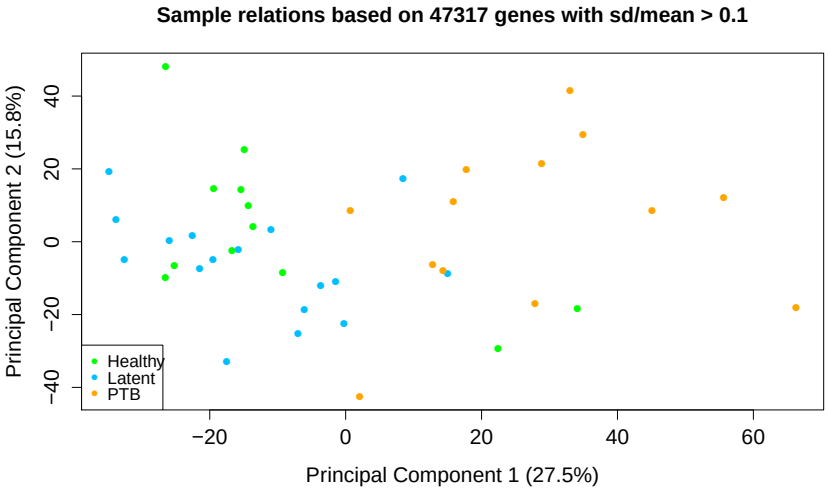
Figure 7.1: **Box-and-whisker plots of the raw expression data for the three datasets.** Colours indicate sample classes: green=*healthy*, blue=*LTBI*, orange=*PTB* and red=*TB-PC*. Overall expression patterns look similar between the three datasets, and no extreme outliers are obvious. Boxes indicate lower quartile, median and upper quartile, and whiskers the extremes of data.



(a) **IMPI-MA**



(b) **BERRY validation**



(c) **BERRY training**

Figure 7.2: **Multidimensional scaling analysis.** The three datasets are shown following multidimensional scaling of the raw data.

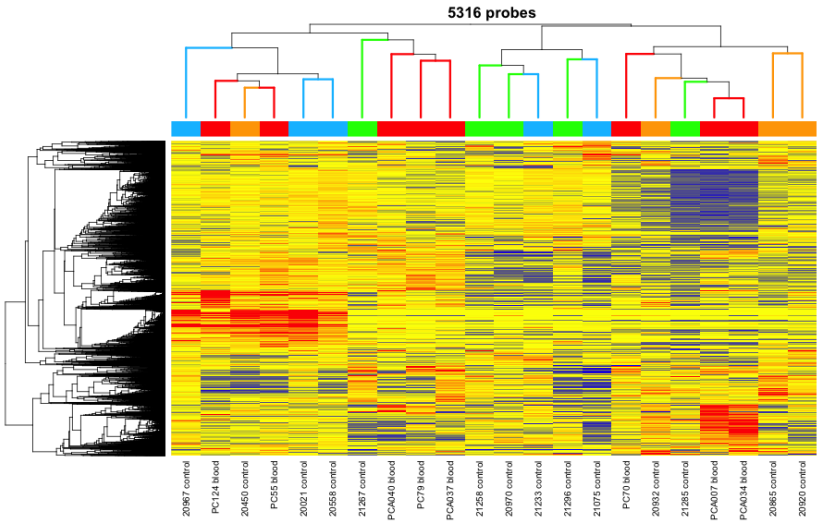
mediate to the other two datasets, and that overall the number of selected probes are closer to the BERRY training set value. The two datasets based on samples collected in South Africa appear to be more variable than the original BERRY training set; the reason for this is unclear.

Figure 7.3 shows heatmaps of the selected probes. The BERRY validation set is already well separated into active TB and LTBI clusters, whereas the other datasets still show high levels of misclassification. The heatmaps in question demonstrate that by only using non-specific filtering to remove non-informative probes, much variability remains in the data obscuring potential differences that may exist between probes belonging to different phenotypic classes. Essentially this shows that in an unsupervised analysis, the various phenotypic groups do not cluster together in the IMPI-MA dataset. This could be due to smaller sample size, or additional variability between samples not explained by disease phenotype. This could include for instance concurrent viral infections in otherwise healthy individuals.

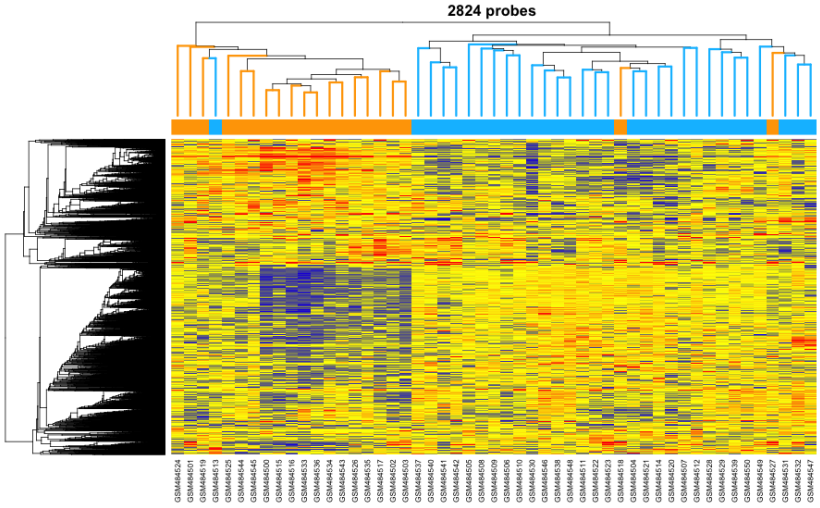
Multidimensional scaling applied to the selected probes (i.e. the probes passing both filters) show separation into clusters that does not relate well to the clinical phenotype.

Differential expression analysis was performed using the Kruskal Wallis test (IMPI-MA, BERRY validation) for the contrast active TB vs. not active TB, and Kruskal Wallis ANOVA for BERRY training set, as in the original paper. All results were corrected for multiple testing using the Benjamini Hochberg procedure. Table 7.3 list the results. This comparison is problematic at multiple levels. First, the number of conditions used for the comparisons is not identical for the three analyses. The original BERRY training set used Kruskal Wallis ANOVA to distinguish between three conditions (healthy, LTBI and PTB). This yields 404 DE probes at an FDR of 0.01, a result that is very similar to the original paper. Two conditions were used for the two other sets, as the validation set did not include healthy controls, and the IMPI-MA subset was too small to yield significant results for three conditions. Also of note is the fact that very different levels of FDR were used for the three analyses. This was done to obtain a roughly similar number of DE probes, even though the proportion of false positives in each results set was likely to be different due to this.

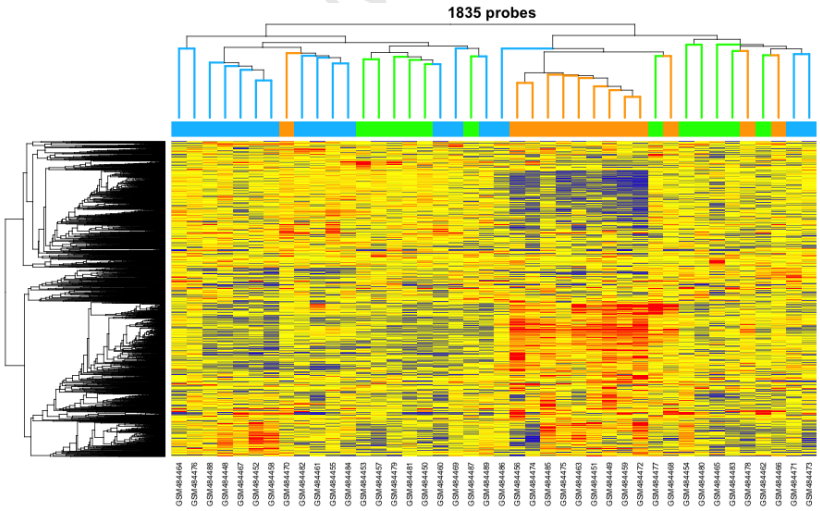
Given these results, and the caveats that accompany them, we now focus on the probes. The differentially expressed probes are shown in heatmaps in Figure 7.5 and multidimensional scaling plots in Figure 7.6. Of interest is whether the selected probes classify their input data correctly. As can be seen in Figure 7.5, the respective lists of differentially expressed probes classify the input



(a) **IMPI-MA**

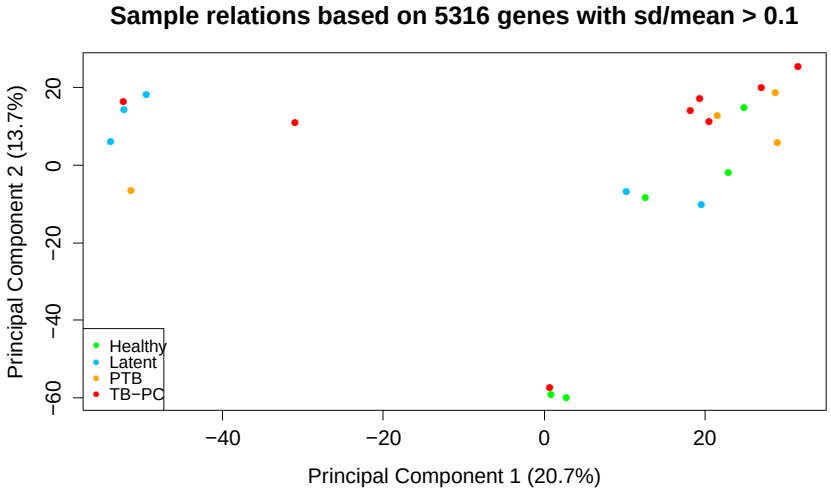


(b) **BERRY validation**

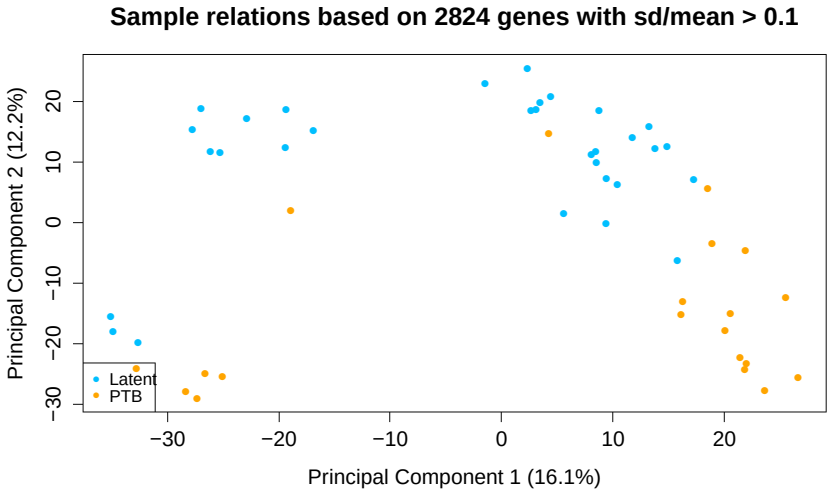


(c) **BERRY training**

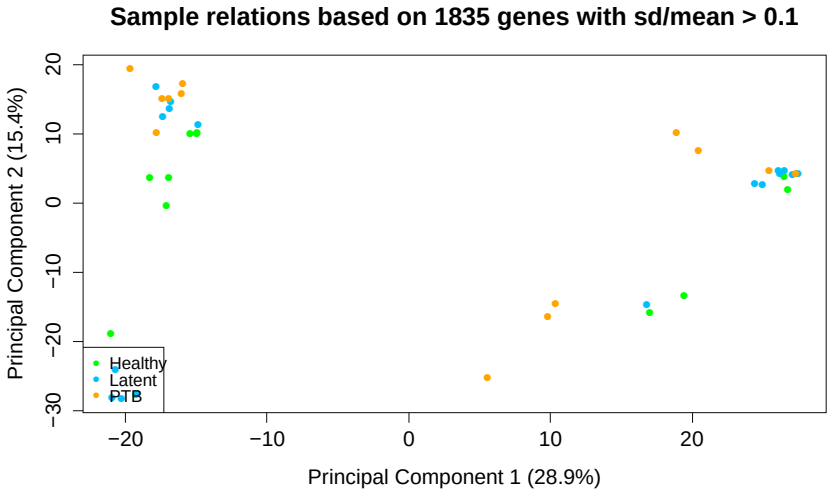
Figure 7.3: **Heatmaps of selected probes.** Each heatmap shows the expression values of the selected probes. Values are row-scaled, and range from low (blue) to intermediate (yellow) to high (red). Samples are clustered using Spearman rank correlation, and probes are clustered using Pearson correlation. Colours: green: *healthy*, blue: *LTBI*, orange: *PTB*, red: *TB-PC*. Plot titles reflect the number of included probes.



(a) **IMPI-MA**



(b) **BERRY validation**



(c) **BERRY training**

Figure 7.4: **Multidimensional scaling analysis of selected probes.**

Table 7.3: **Results of differential expression analysis**

Dataset	N conditions	FDR	N DE probes
IMPI-MA	2	0.08	394
BERRY validation	2	0.0001	484
BERRY training	3	0.01	404

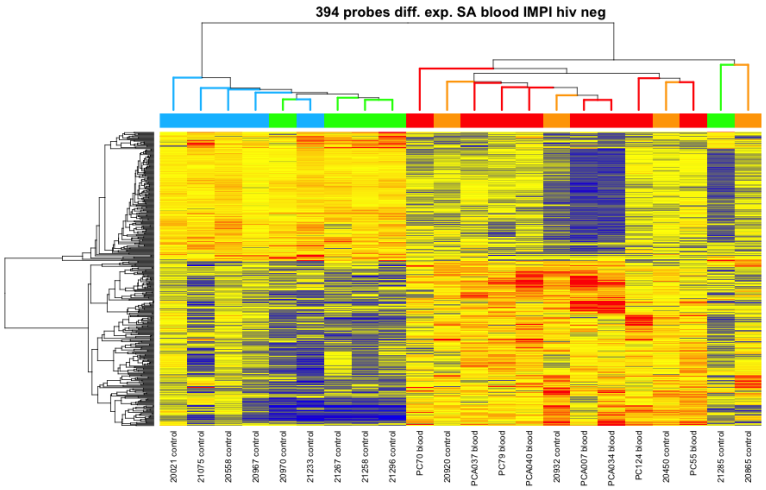
data reasonably well. In Figure 7.5a we find that one sample ("healthy" phenotype) is misclassified in the IMPI-MA data; in case of the BERRY validation data, two samples are misclassified (one "active TB" and one "LTBI" phenotype). Finally, in the case of the BERRY training set, three samples (one each of "healthy", "LTBI" and "active TB") are misclassified.

7.4 Comparisons

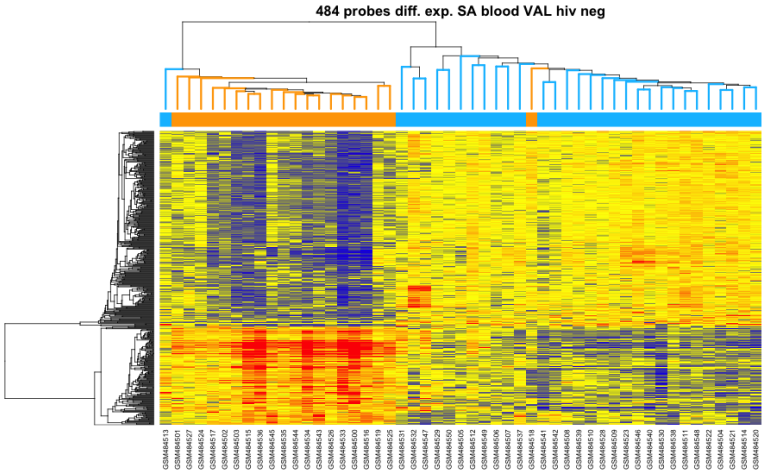
The main objective for the Berry et al analysis was to derive a predictive signature, whereas the main objective for the analysis in this chapter is to assess, given independent input data, how robustly does the method employed in the Berry et al analysis produce a characteristic signature, and how do signatures produced overlap with the original 393 signature? We can regard this approach as repeating the training set analysis three times, and comparing the results.

Firstly, how does the implementation of the Berry et al analysis method in R compare to the original results? Figure 7.7 shows a two-way Venn diagram demonstrating the overlap. One would expect perfect overlap if the method is indeed identical. We see that all probes from the 393 signature are reproduced by the R implementation of the method. An additional eleven probes are also called significant; this may be due to rounding errors due to machine precision settings used on two different systems; this difference may be caused by inclusion of probes at either of the non-specific filters or at the time of DE analysis using Kruskal Wallis ANOVA. In any event, the R implementation of the gene selection procedure reproducibly recreates the 393 probe list with a few additional probes.

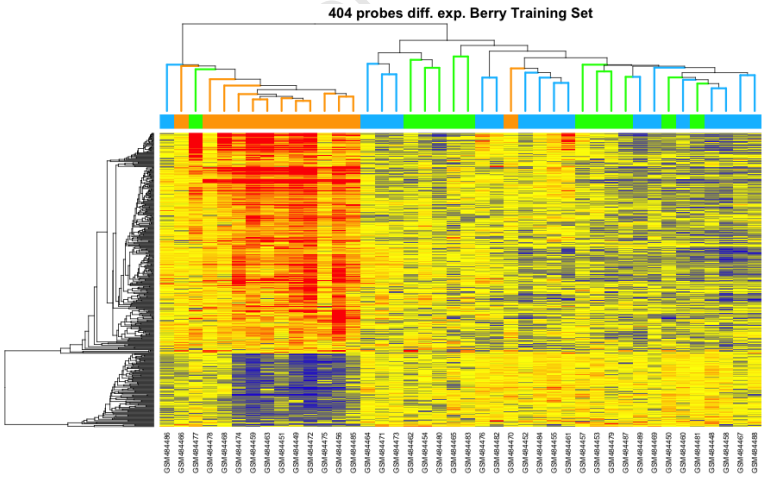
The second question we now address is how do the probes selected from the IMPI-MA data overlap with the original 393 signature. Figure 7.8 shows this overlap. Interpretation of this extent of overlap requires some context before we proceed. In the *Gedankenexperiment* ("thought experiment") that follows, we consider two sets A and B of 30,000 unique elements, and ask the question:



(a) **IMPI-MA**

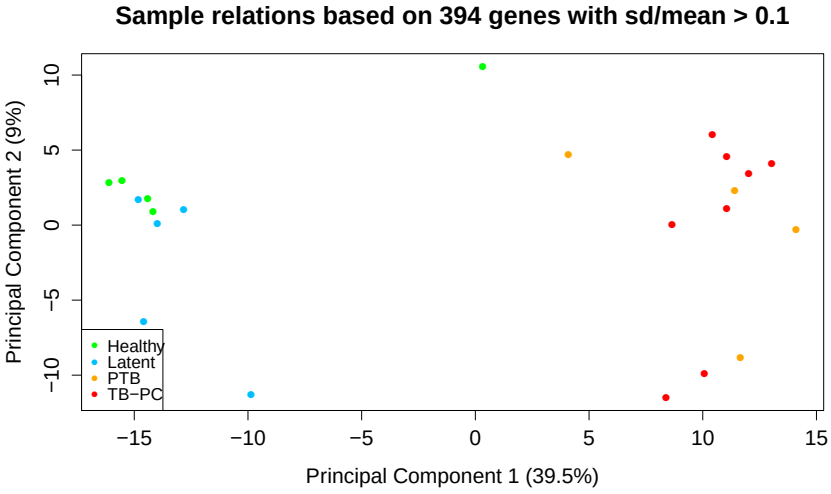


(b) **BERRY validation**

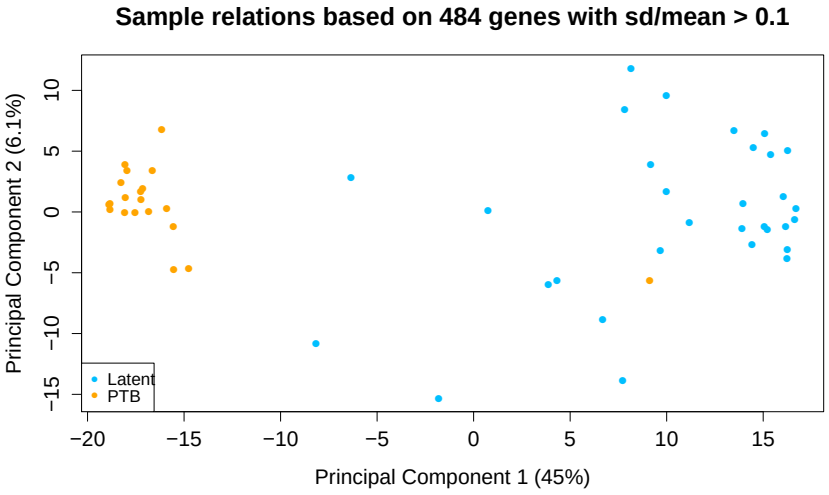


(c) **BERRY training**

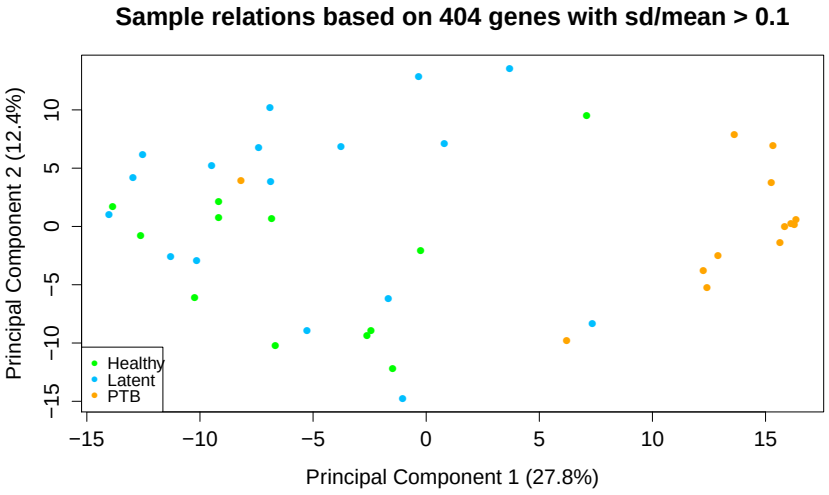
Figure 7.5: **Heatmaps of differentially expressed probes.** Each heatmap shows the expression values of the selected probes. Values are row-scaled, and range from low (blue) to intermediate (yellow) to high (red). Samples are clustered using Spearman rank correlation, and probes are clustered using Pearson correlation. Colours: green: *healthy*, blue: *LTBI*, orange: *PTB*, red: *TB-PC*. Plot titles reflect the number of included probes.



(a) **IMPI-MA**



(b) **BERRY validation**



(c) **BERRY training**

Figure 7.6: **Multidimensional scaling analysis of differentially expressed probes.**

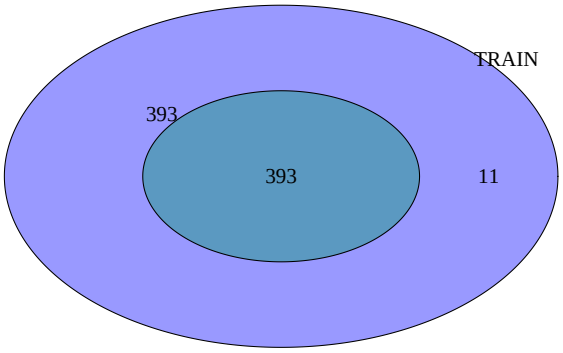


Figure 7.7: **Method validation.** Overlap of "393" signature with DE probes from"BERRY training set ("TRAIN") as found using R. Each set is labelled on the circle boundary, and the venn counts are centred within each circle. Please note that the *oval sizes are not scaled* to the number of probes for the purpose of better legibility.

how probable is an overlap of size n of two subsets of A and B, each of size k? Using *Mathematica,* I created two sets A and B, each consisting of 30,000 elements, randomly subset each by 400 elements, and calculated the overlap (strictly the size of the intersection of the two subsets $A_k \cap B_k$). This was repeated 20,000 times, and the resulting data charted as a histogram (Figure 7.9). I then fit three types of distribution to the data (a normal distribution, a kernel mixture distribution and a Pareto distribution (Figure 7.9)). Using this distribution, I then produced a probability plot for different sizes n; this shows that any n>15 is very unlikely, with a probability of 0.000655873. An overlap size exceeding 100 is extremely unlikely to occur by chance (probability=$2.59115 * 10^{-176}$). I conclude that most probes in the overlapping region are present because of biological reasons. The non-overlapping probes may be false positives and false negatives, respectively, depending which dataset is used as reference.

The two datasets were generated using different versions of the array technology, consisted of different sample sizes, were drawn from different populations and were analysed using different values for false discovery rate. Despite this, a statistically highly significant overlap of differentially expressed probes gives me confidence that overall, the IMPI-MA data is comparable to the Berry et al data.

Finally, we show the overlap of the 393 probe set with the BERRY training set results using R, the BERRY validation set and the IMPI-MA data in Figure 7.10.

Despite the fact that this degree of overlap between three datasets is extremely unlikely to occur by chance, we find that 61 probes are present as differentially expressed probes in the BERRY

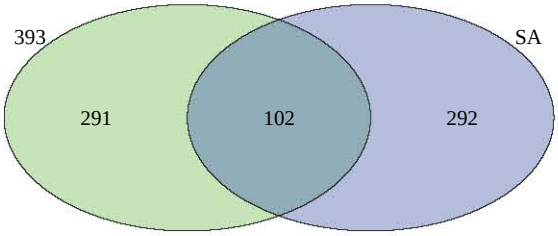
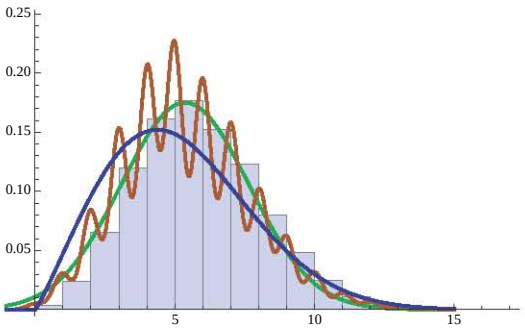
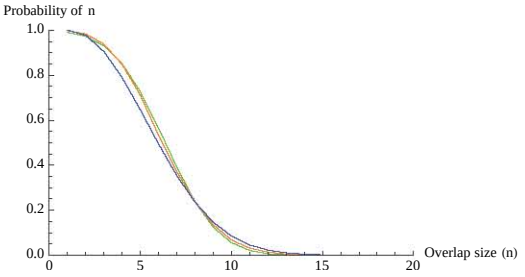


Figure 7.8: **Data similarity.** Overlap of 393 probe set and the DE probes selected from the IMPIMA data using the same analytic method from a dataset that differs in many respects from the BERRY training set.



(a) **Overlap size histogram**. Histogram of overlap sizes after 20,000 sampling runs, overlaid by the PDF of three distributions (Colours: green, normal distribution; orange, kernel mixture distribution; blue, Pareto distribution)



(b) **Probability plot for three distributions.** (Colours: green, normal distribution; orange, kernel mixture distribution; blue, Pareto distribution).

Figure 7.9: **Probability of probe overlap.**

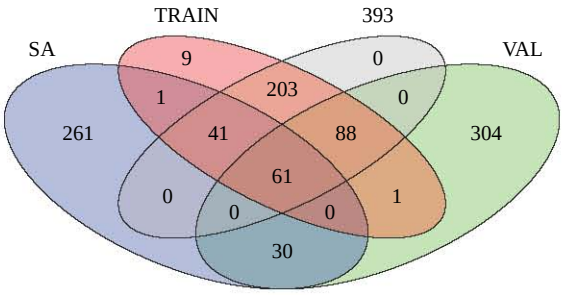
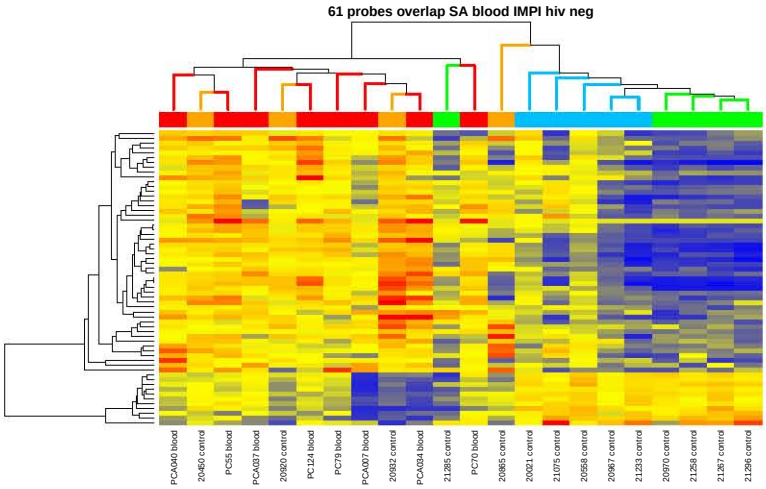


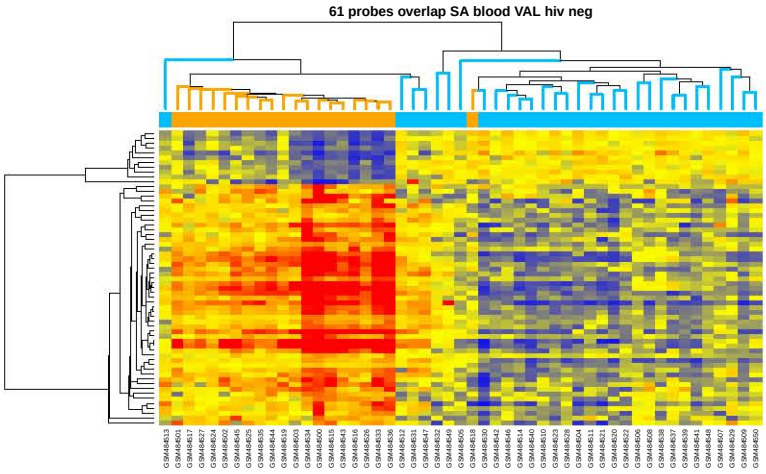
Figure 7.10: **Overlap of all three datasets with the 393 probe list**

training set, validation set and IMPI-MA subset. I infer that these probes are particularly interesting from a biological point of view. Do these probes allow for clear separation of active TB and not active TB? To investigate this, I show in Figure 7.11 three heatmaps, each consisting of the 61 probes present in all result sets, and demonstrate that many samples are still correctly classified as active or not active TB. Of note is that assignment to either the active or not active TB cluster is now more frequently done incorrectly. In the original analysis, the samples of IMPI-MA, BERRY validation and BERRY training sets are misclassified in one, two, and three instances, respectively, while using the 61 set of overlapping probes results in two, five and eight samples misclassified. However, active and not-active TB samples still cluster together in each of the two main clusters.

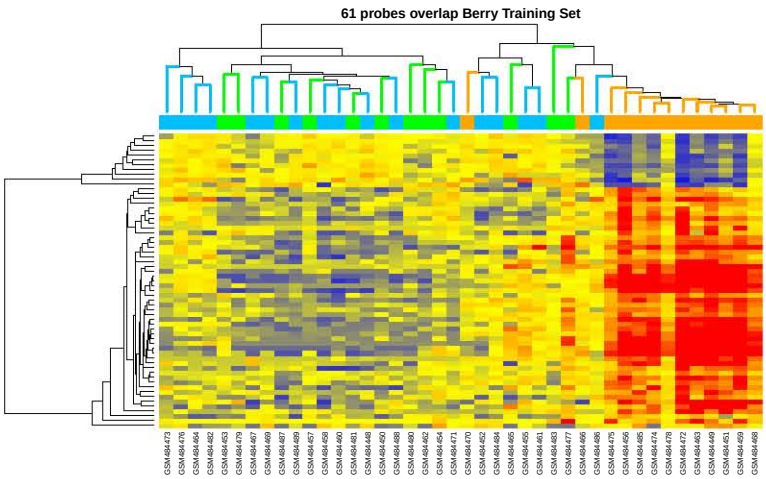
While the 61 probe subset does not appear to improve the classification of the input data, further investigation of the actual probes that constitute the set will be required to determine their biological significance. For instance, the probes may capture a conserved feature of the immune response to *M. tuberculosis* detectable in blood that is independent of site or severity of disease.



(a) **IMPI-MA**



(b) **BERRY validation set**



(c) **BERRY training set**

Figure 7.11: **Heatmaps of 61 overlapping probes applied to three datasets.** Each heatmap shows the expression values of the selected probes. Values are row-scaled, and range from low (blue) to intermediate (yellow) to high (red). Samples are clustered using Spearman rank correlation, and probes are clustered using Pearson correlation. Colours: green: *healthy*, blue: *LTBI*, orange: *PTB*, red: *TB-PC*. Plot titles reflect the number of included probes.

8 An analytic framework for heterogenous microarray data

Chapter summary

In this chapter I develop the idea of the data in relation to a multi-dimensional hypercube further, and show how this was used to develop an analytic framework for the IMPI-MA data. Finally, the code structure for this analytic framework is described, and an overview of the output of the analysis is given.

8.1 The sample hypercube revisited: what kinds of questions relating to the three main hypotheses can we ask, and how can we answer these questions?

Having verified that the data are sound (Chapter 6) and comparable to previously published data (Chapter 7), we now turn our attention to the addressing the three main hypotheses stated earlier.

8.1.1 Questions

Given the three hypotheses, the questions we aim to ask are related to the detection of differential transcript abundance for three main contrasts: TB status, HIV status and compartment. Each of these questions may be posed in different contexts, e.g. differential gene expression (DE) in active TB vs. not active TB in HIV-1 infected and HIV-1 uninfected individuals, respectively. The results of DE experiments may then be compared between these contexts, allowing for assessment of the effect of HIV-1 infection on the transcriptional response to tuberculosis. Table 8.1 lists the

questions and contexts developed for the analysis of the IMPI-MA data, as well as various metadata variables that will be required to subset the main dataset, set up the contrasts and specify annotation variables, if different from the contrast variable. Figure 8.1 shows the questions (1-7) mapped to 3D phenotype space, represented as the hypercube graph discussed earlier.

Table 8.1: **Questions.** Seven questions addressed in IMPI-MA analysis, listing metadata relevant to the implementation in R

Question	1: Tuberculosis	2: TB site	3: LTBI	4: HD phenotype
Contrast variable	class3	class	class	ECP
Contrast 1	ActiveTB	PTB	Healthy	Eff TB-PC
Contrast 2	notActive TB	TB-PC	LTBI	EC TB-PC
Colour variable	class	class	class	ECP
Data	B.Neg.AN	B.Pos.PTB_TBPC	B.Pos.HL	B.C.Pos.EffEC
	B.Pos.AN	B.Neg.PTB_TBPC	B.Neg.HL	PF.C.Pos.EffEC

Question	5: HIV (not active TB)	6: HIV (active TB)	7: Compartment
Contrast variable	HIV.Status	HIV.Status	Compartment
Contrast 1	Positive	Positive	Blood
Contrast 2	Negative	Negative	Fluid
Colour variable	HIV.Status	HIV.Status	Compartment
Data	B.N.PosNeg	B.A.PosNeg	C.BPF (matched)
	B.L.PosNeg	PF.A.PosNeg	C.Neg.matchedBPF
	B.H.PosNeg	B.PTB.PosNeg	C.Pos.matchedBPF
		B.TBPC.PosNg	C.Pos.EFF.matchedBPF
			C.Pos.EC.matchedBPF

Datasets are subsets of all IMPI-MA data, named in triads or tetrads using . as separator. Each triad (tetrad) specifies SET.SUBSET.(SUBSET).CONTRAST. Abbreviations for datasets: **B**=blood, **PF**=pericardial fluid, **Neg**=HIV-1 uninfected, **Pos**=HIV-1 infected, **A**=active TB, **N**=not active TB, **L**=LTBI, **H**=healthy, **PTB**=pulmonary tuberculosis, **TBPC**=tuberculous pericarditis, **C**=tuberculous pericarditis (cardiac), **Eff**=effusive pericarditis, **EC**=effusive-constrictive pericarditis. Contrast variable refers to the name of the variable in the clinical phenotype dataset used to define the contrast of interest. Colour variable refers to the name of the variable in the clinical phenotype dataset used to define the factor used for identifying subsets in graphics by colour.

8.1.2 Analyses

Each dataset listed in Table 8.1 was analysed by the three complementary approaches described in Methods 3.8.

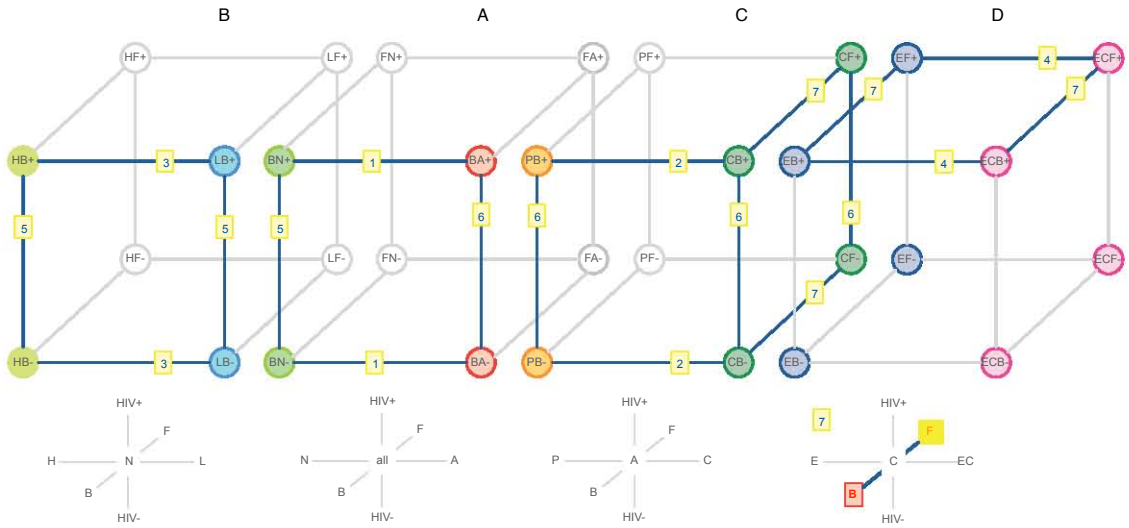


Figure 8.1: **Sample hypercube**. Questions listed in Table 8.1 mapped to edges of the six-dimensional phenotype space rendered in three dimensions. Yellow labels indicate the question number; labelled edges represent the contrasts examined; greyed-out edges and vertices represent contrasts and sample classes not available or inappropriate for analysis. **Cubes**: **A** *Main cube* (Tuberculosis, HIV-1 infection, Compartment) **B** *Not active tuberculosis* **C** *Active tuberculosis* **D** *Tuberculous pericarditis* **Abbreviations**: **B** blood; **F** pericardial fluid; **A** active tuberculosis; **N** not active tuberculosis; **+** HIV-1 infected; **-** HIV-1 uninfected; **m** matched samples exist

8.2 Code description

8.2.1 IMPI-MA: Data manager

The code for this analysis is reproduced in full in Appendix Section A.11. Using Table 8.1 as reference, the script is based on a similar script described earlier (A.7), but extracts all data subsets based on the vertices of the extended hypercube and the selected contrasts and contexts. This data is saved to RData files for later access.

8.2.2 IMPI-MA: Analysis

The code for this analysis is reproduced in full in Appendix Section A.12. Structurally, the code implements a for-loop, wherein each of the seven questions is addressed. Within each question, one or several datasets are each subjected to the same analytic workflow of differential expression, deconvolution and cell-specific differential expression, and weighted gene co-expression network analysis. Multiple mechanisms for dealing with code exceptions have been implemented. These deal with missing or incomplete data and insufficient sample size. In addition, memory management has been optimised, allowing the code to run with a remarkably small memory footprint. A single run of this script takes approximately three and a half hours to complete on a 2010 intel dual core i7 machine with 8GB of non-ECC RAM running Mac OSX 10.8.4.

An important feature of this code is that the output is written to disk using versioned folders, and the naming of individual output files is based on the question, analysis type, dataset and contrast and output type. This fully automated process generates output files where the file name contains a significant amount of metadata allowing for interpretation of the output without reference to external material.

Outer for-loop

The outer for-loop iterates over a list containing seven “questions”. Each question in turn is a list, containing all metadata required to perform the analyses and correctly label all output. The overall code structure is shown in Figure 8.2.

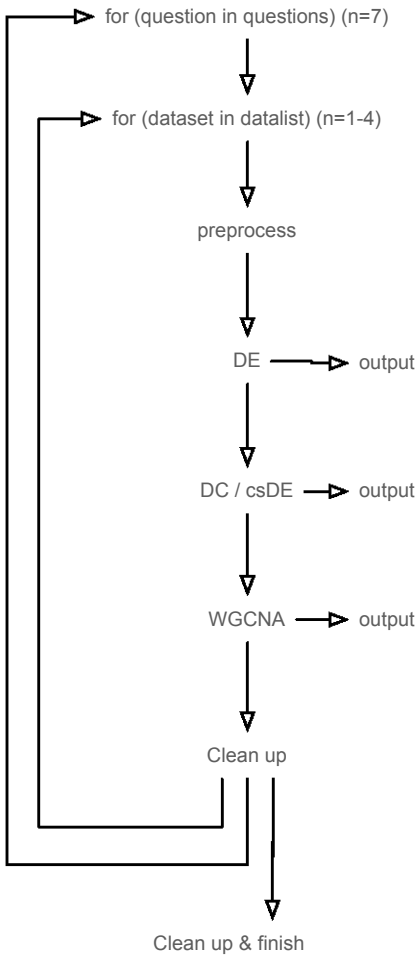


Figure 8.2: **Overall code structure for IMPI-MA main script**

Generation of clinical characteristics tables An important consideration for microarray analysis of clinical samples is the description of the clinical characteristics of the patients included in the particular analysis. This is required in order to identify potential confounding effects, and to provide confidence that the two groups identified by the contrast variable are similar, and differ only in the value of the contrast variable. This essentially means that the two groups to be contrasted have a Hamming distance of one. In addition, where an analysis of the same contrast is performed in more than one dataset (i.e. context), it is also useful to show how the different datasets vary according to clinical parameters.

Given the complexity of this task, the following phenotypic variables were chosen for the clinical characteristics tables: age, sex and CD4 count. Each clinical table compares these variables within datasets (across contrasts) and between datasets (across contexts).

The comparisons for each variable are computed and stored in temporary variables. The results of the various computations are then combined into a single table and exported to raw LaTeX.

Differential gene expression

The DE section of the script performs the following actions:

1. Show raw data (boxplots, MDS, PCA)
2. Preprocess the data: variance stabilising transform followed by quantile normalisation
3. Show pre-processed data
4. Apply filter to remove poor-quality probes
5. Show filtered probes
6. Perform differential expression analysis
7. Show DE results
8. Export the probe lists

Deconvolution and cell-specific differential expression

The deconvolution section of the script performs the following actions:

1. Perform deconvolution based on method in [100]
2. Show proportion matrix as stacked barplot
3. Show boxplots of cell-type proportions by condition
4. Calculate statistics to identify cell-types that differ in proportion by condition (includes multiple testing correction)
5. Perform cell-type specific differential expression analysis using csSAM [135] and the proportion matrices calculated in 1.
6. Generate top tables for each cell type and export

Weighted gene co-expression network analysis

The weighted gene co-expression network analysis performs the following actions:

1. Setup and linking of clinical trait data to expression data
2. Block-wise network construction and module detection
3. Relate modules to external information and identify important genes
4. Output module gene lists and perform gene-ontology analysis
5. Visualise networks of module eigengenes in relation to clinical traits
6. Export module networks in Cytoscape format
7. Plot module heatmaps and eigengene expression
8. Plot boxplots of eigengene expression values by condition (contrast)
9. Calculate statistics to identify modules that differ in expression by condition (includes multiple testing correction)

8.3 Computational output

8.3.1 Output tree

All output is saved in a single directory named using the date and start time of the run in R that produced the output. Two folders are made within the output folder; one contains all images, and the other all text files.

8.3.2 Images

Images can be output in multiple raster or vector formats; in order to conserve disk space all images were output as pdf for the run that produced the output for Chapters 9 to 12. This may be changed by modifying parameters in the setup script. A single run outputs 2,182 images (at the time of writing). All image files are labeled automatically with four numbers and a text string in the format "question#", "analysis#", "dataset#", "figure#" followed by text that specifies the question, analysis, dataset and type of output. This allows anyone to understand a particular output file simply by inspecting the filename. In addition, most plots contain some annotation on the image (usually the image title) that allows the viewer to understand what (s)he is looking at.

8.3.3 Session Output log

All output usually written to *stdout*, i.e. the R console under usual configurations of R, is captured in a text file entitled *sessionOutput*.txt. This text file captures all warning or error messages created while running the code, as well as the output of print statements and any output normally sent to *stdout*. This file serves two main purposes: (1) Debugging the script by catching informative error messages and (2) capturing all output not written to file.

The file produced in a single run of the main analysis script is 84 pages long when printed. Table 10.2 and other similar tables were produced based on output in this file. For such a comprehensive analysis, this type of output serves as a useful reference point when evaluating the overall results of an analysis run.

8.3.4 Spreadsheets

314 spreadsheet-like objects were produced during the course of the analysis. These objects are comma-delimited text files and can easily viewed in any spreadsheet programme. The following analytic steps produce these objects:

1. Differential expression analysis
2. Cell-type specific differential expression analysis (one sheet for each cell type)
3. Gene Ontology enrichment analysis, as produced during WGCNA analysis
4. Results of statistical calculations (e.g. evaluating statistical significance of of differential module regulation or differences in cell-type proportions between sample classes)
5. Results of gene significance and module membership calculations

8.3.5 Plain text files

1373 plain text files were produced in the course of the analysis. They are simpler in structure than the csv files, and contain the following classes of data:

1. Entrez IDs for all genes used in the WGCNA analysis
2. Entrez IDs for all genes in modules identified in the WGCNA analysis

3. Nodes and edges for all modules in weighted gene correlation networks for all analyses. These files can be imported into Cytoscape (a network visualisation and analysis tool) for further processing.

8.3.6 LaTeX tables

Tabular data for display purposes was formatted in the script as LaTeX tables, and written to tex files. These files could then be incorporated directly in documents like this thesis. Table 10.1 and other similar tables were produced in this way. A single run of the script at the time of writing produces 27 ready to use LaTeX documents.

8.4 Results as Data: a new resource for generating and analysing HIV-TB related hypotheses

I have shown that the output of a complex analytic pipeline may yield very comprehensive output that requires further systematic evaluation in order to understand fully. In this way, one output described in this thesis is a large dataset consisting of multiple output files that as a whole addresses a range of questions. In this way, the results of running the analytic pipeline produces new classes of data that are amenable to further in-depth analysis.

Part III

IMPI-MA: Results in depth

9 Question 1: Tuberculosis

Chapter summary

Matthew Berry et al have previously published a transcriptomic signature for tuberculosis in HIV uninfected individuals [82], and others have since confirmed many properties of the signature that was found [86, 85]. Here I show the results of an analysis that refines and extends the work of these authors. I present the results analysis for the dataset of Question 1: ***Blood, HIV negative*** with the contrast ***activeTB vs not active TB*** as an example of the depth of the analytic framework and the associated output. As mentioned in Chapter 8, the entire analysis output is extremely comprehensive, and will not be reproduced in full for the purposes of this thesis. This chapter is divided into three parts, presenting results for differential expression analysis, deconvolution and cell-specific differential expression analysis, and weighted gene co-expression network analysis. Pathway analysis will be presented in the next Chapter.

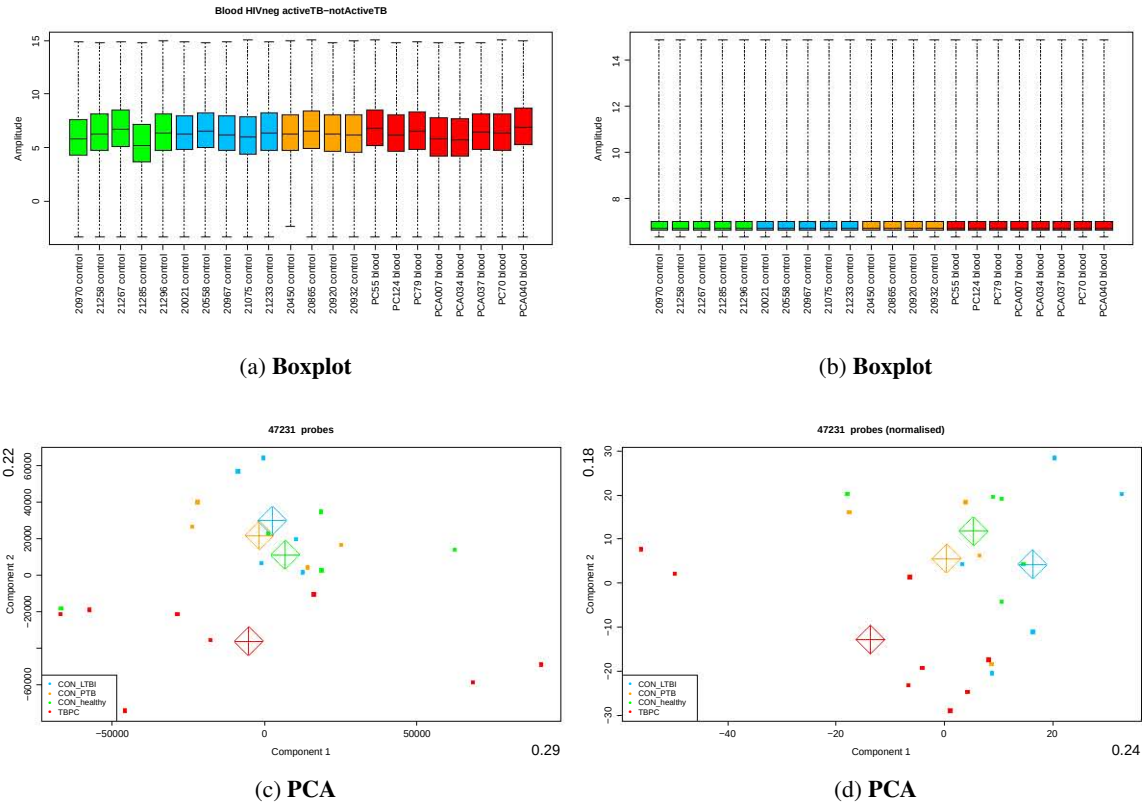
9.1 Differential gene expression

In this section I present the results of differential expression analysis using quantile normalised data and a linear models approach. The contrast used was *active TB vs. not active TB*. *Active TB* includes **PTB** and **TB-PC,** while *not active TB* includes **healthy** and **LTBI** classes.

9.1.1 Data quality

Boxplots of the non-normalised data show that no sample differs significantly from all others (Figure 9.1) and the quantile normalised data distributions for each sample are identical. PCA analysis of the non-normalised as well as normalised data demonstrate no significant batch effects; there is

large overlap of the four groups displayed.



(a) **Boxplot** (b) **Boxplot**

(c) **PCA** (d) **PCA**

Figure 9.1: **Data QC: not normalised (left) and normalised (right).**

9.1.2 DE probes

Following normalisation and filtering of probes based on probe quality, differential expression analysis was performed using the R package *limma*, with false discovery rate set to 0.05. 369 probes were selected as differentially expressed. Table 9.1 lists the top 30 differentially expressed probes, ordered by the variable *logFC* ($\log_2$-fold change). A positive value indicates overexpression in active tuberculosis, and a negative value underexpression in active tuberculosis.

The results of the differential expression analysis can be visualised using a volcano plot (Figure 9.2), which, for each probe, plots the $\log_2$-fold change value against the log-odds for differential expression (***B***). We conclude from the plot that the majority of probes are unlikely to be differentially expressed, and have low logFC values. The significant probes are labeled by identifier in blue.

Table 9.1: **Top 30 probes as selected by *limma***

ProbeID	TargetID	logFC	CI.025	CI.975	AveExpr	t	P.Value	adj.P.Val	B
4150270	ANKRD22	1.77	1.23	2.31	8.78	6.41	0.00	0.00	4.98
6060468	S100A8	1.75	1.26	2.23	12.69	7.01	0.00	0.00	6.18
130181	ANKRD22	1.56	1.06	2.07	8.21	6.12	0.00	0.01	4.39
6620209	FCGR1B	1.32	0.98	1.65	7.95	7.75	0.00	0.00	7.58
520360	MS4A6A	1.05	0.72	1.38	10.06	6.27	0.00	0.00	4.69
580706	CCR2	1.02	0.72	1.33	8.86	6.56	0.00	0.00	5.30
840168	CYBB	0.99	0.73	1.26	9.24	7.36	0.00	0.00	6.85
6040711	CCR2	0.98	0.68	1.27	8.72	6.50	0.00	0.00	5.17
450491	CASP1	0.89	0.62	1.15	9.36	6.51	0.00	0.00	5.20
6560156	DUSP3	0.88	0.59	1.16	9.98	6.03	0.00	0.01	4.20
7050382	CASP1	0.86	0.60	1.13	9.44	6.32	0.00	0.00	4.82
20452	TYMP	0.85	0.58	1.12	8.54	6.24	0.00	0.00	4.64
5570039	LOC728744	0.75	0.57	0.92	7.34	8.28	0.00	0.00	8.52
1770152	MS4A6A	0.74	0.51	0.97	10.12	6.38	0.00	0.00	4.93
830035	ATG3	0.72	0.49	0.94	9.37	6.26	0.00	0.00	4.68
240053	GCH1	0.63	0.44	0.83	7.64	6.39	0.00	0.00	4.95
4040022	CD86	0.58	0.40	0.76	7.98	6.40	0.00	0.00	4.97
2360719	IRAK3	0.54	0.37	0.72	7.84	6.09	0.00	0.01	4.33
7200753	TLR7	0.51	0.35	0.66	7.59	6.53	0.00	0.00	5.24
870408	IL15	0.48	0.32	0.63	7.35	6.04	0.00	0.01	4.22
670202	SESTD1	0.47	0.32	0.62	7.71	5.99	0.00	0.01	4.13
1660615	CCR2	0.20	0.14	0.25	6.88	7.02	0.00	0.00	6.20
7050328	SEMA4C	-0.16	-0.22	-0.11	6.97	-6.03	0.00	0.01	4.20
5700189	TCTN1	-0.27	-0.36	-0.18	7.22	-6.03	0.00	0.01	4.21
580411	LAX1	-0.46	-0.59	-0.32	7.94	-6.72	0.00	0.00	5.61
5260021	HS.135282	-0.46	-0.61	-0.32	7.47	-6.46	0.00	0.00	5.08
7550008	RNF216	-0.55	-0.70	-0.39	8.10	-6.99	0.00	0.00	6.15
3940484	MEF2D	-0.58	-0.75	-0.42	7.98	-6.87	0.00	0.00	5.92
130609	FCGBP	-0.83	-1.08	-0.58	7.82	-6.44	0.00	0.00	5.05
4010397	PLEKHG3	-0.95	-1.26	-0.64	9.10	-6.06	0.00	0.01	4.26

Key to variables: ***ProbeID***: Illumina probe identifier, ***TargetID***: Gene name, ***logFC***: $\log_2$-fold change, ***CI***: lower and upper bounds of 95% confidence interval for logFC variable, ***AveExpr***: average expression value for all arrays, ***t***: t-statistic, ***P.Value***: raw P-value for linear model coefficient, ***adj.P.Val***: Benjamini-Hochberg corrected P value for linear model coefficient, ***B***: log-odds for differential expression.

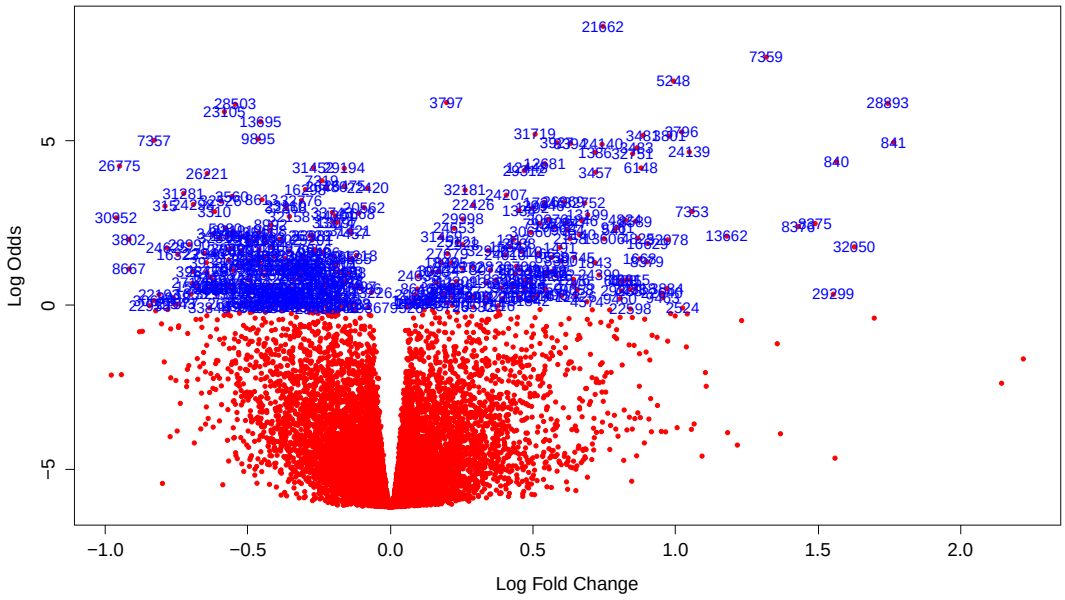
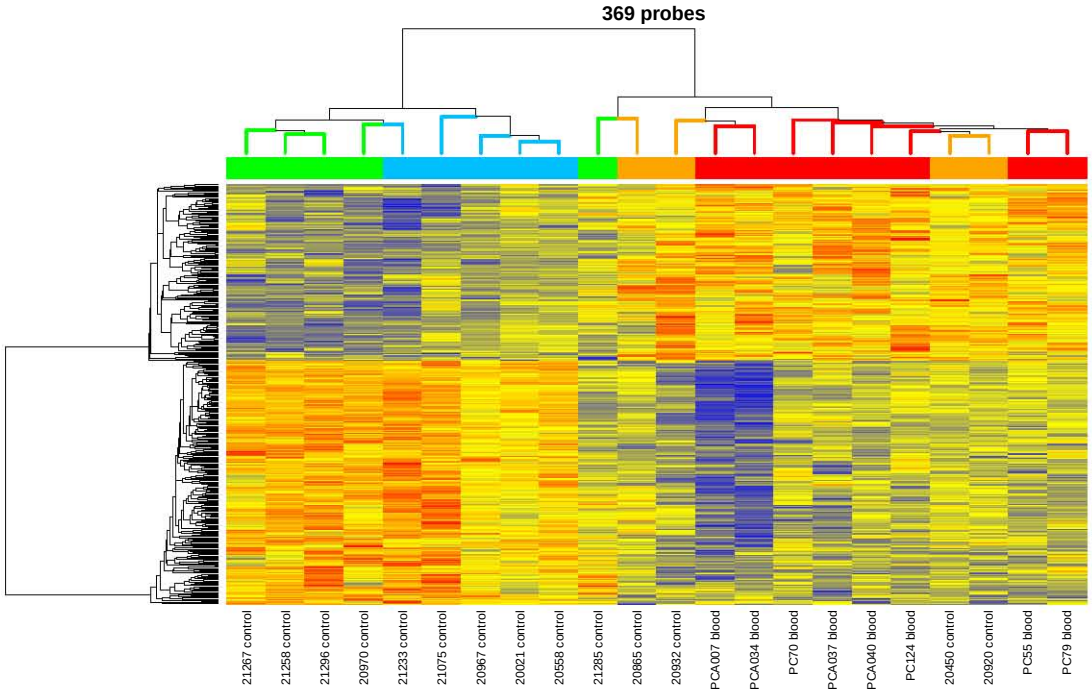


Figure 9.2: **Volcano plot of differentially expressed probes.** The significant probes are labeled by identifier in blue.

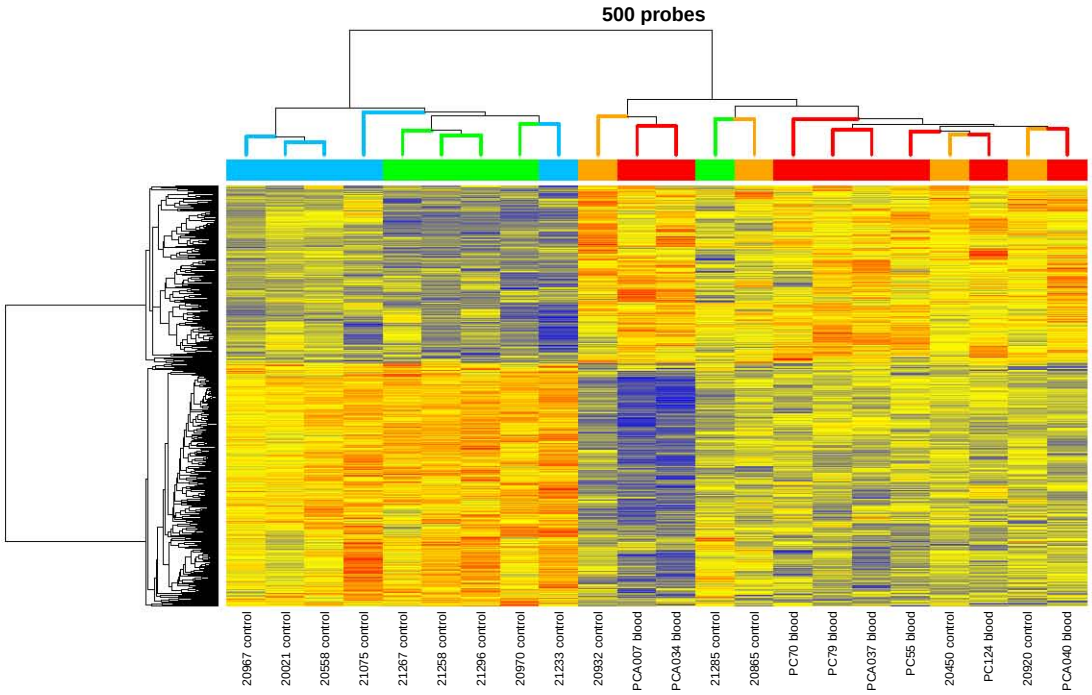
A more informative plot shows the expression values for differentially expressed probes as a heatmap, with hierarchical clustering of both samples and probes. Heatmaps of the input data subset by significant or top 500 probes are shown in Figure 9.3. In both cases, clustering of samples is near-perfect, with one sample (not active TB) assigned incorrectly to the active TB cluster. It is further interesting to note that PTB and TB-PC are not separated into distinct subgroups, whereas healthy and LTBI samples are grouped into separate clusters to some extent. We will return to this in Section 10.3.

Finally, Figure 9.5 shows the overlap of selected probes from the same dataset using the method used in Chapter 7 ("Baylor") and the method used throughout the remainder of this thesis ("Limma"). 34% of probes overlap. This shows that the pipeline used for gene selection can have a major impact on the results. Keeping in mind that the normalisation method used by the Baylor method is less stringent, and that a higher FDR rate was used for the Baylor analysis, this should not be too surprising.

Based on the differential expression analysis of this dataset, we conclude that an informative signature for active tuberculosis, including non-pulmonary tuberculosis, has been identified in blood.



(a) **Heatmap of all significant probes**



(b) **Heatmap of top 500 probes**

Figure 9.3: **Heatmaps of significant and top 500 probes** Each heatmap shows the expression values of the selected (a) or top 500 (b) probes. Values are row-scaled, and range from low (blue) to intermediate (yellow) to high (red). Samples are clustered using Spearman rank correlation, and probes are clustered using Pearson correlation. Colours of samples: green: *healthy*, blue: *LTBI*, orange: *PTB*, red: *TB-PC*. Plot titles reflect the number of included probes.

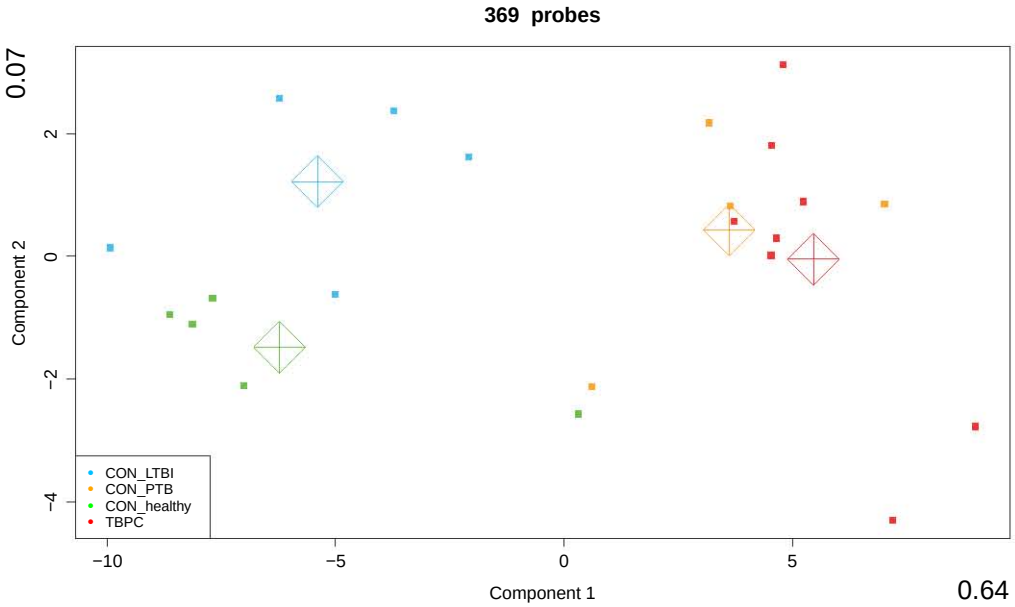


Figure 9.4: **PCA of differentially expressed probes.** Principal component analysis shows that the selected probes clearly separate active tuberculosis from not active tuberculosis

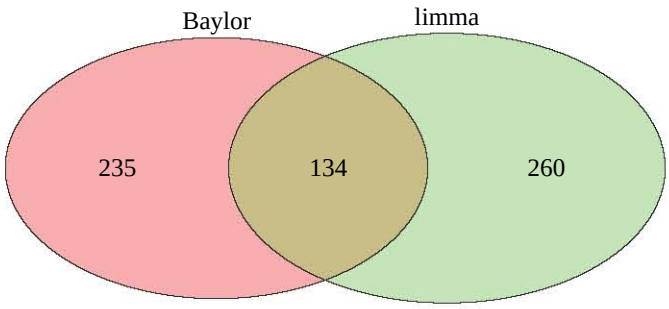


Figure 9.5: **Overlap of differentially expressed probes for two different methods.**

Further work is required to fully characterise the biological meaning of this signature.

9.2 Array deconvolution and cell-specific differential gene expression

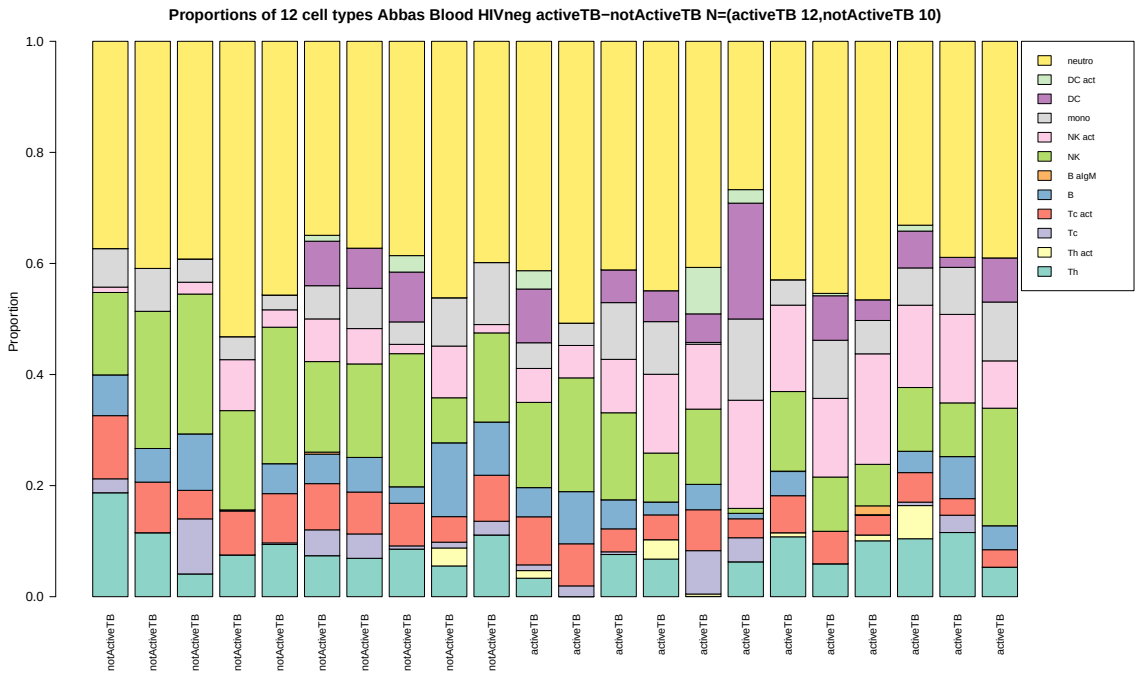
In this section we examine cellular composition of the samples, and the effect that cellular composition may have on differential gene expression.

9.2.1 Deconvolution results

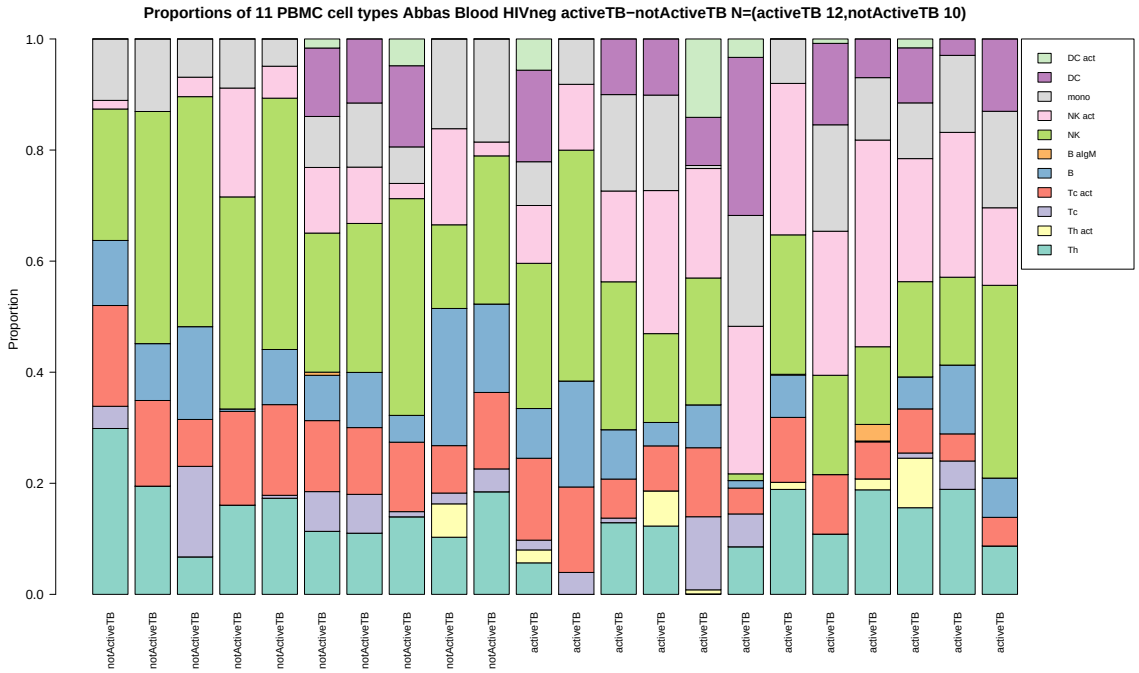
The results of the deconvolution analysis are shown in Figure 9.6. The algorithm, based on the least-squares algorithm described by Abbas et al [100], and implemented in the *CellMix* package [134], returns a matrix of proportions of cell types in each sample. Some important caveats apply: Firstly, the matrix of proportions only includes cell types for which expression profiles were provided. Therefore, even if other cell-types are present in the sample, these will *not* contribute to the proportions matrix. The original paper by Abbas described expression profiles for 18 cell types, and only 12 are reported in the output. This is probably due to the fact that not all probes in the Abbas dataset (Affymetrix data) are directly mappable to the Illumina data. Indeed, of 359 features defining 17 cell-types in the *CellMix* package, 212 are mapped to target IDs, and after removal of duplicate probes, only 143 features are used for the final deconvolution analysis. It is unclear whether an improved probe mapping will substantially alter the results. Future work will re-examine this issue, and also compare these results to other deconvolution methods.

The proportions for each cell-type were then summarised for each group, and the results plotted as pairwise box-and-whisker plots (Figure 9.7). These plots are more informative than the stacked barcharts, as they allow for easy identification of large systematic differences between sample groups. The median proportions for each cell-type were statistically compared between the two disease phenotypes, and the results are presented in Table 9.2.

An interesting and unexpected finding is the highly significant difference in the proportion of natural killer cells (NK cells) that are activated. NK cells are bone-marrow derived mononucleated cells that originate from lymphocyte progenitor cells. In contrast to lymphocytes, NK cells have invariant receptors, and respond to a broad range of intracellular pathogens. In order to respond



(a) **Detected cell types**



(b) **PBMC cell types**

Figure 9.6: **Barplots of detected and PBMC types.** The stacked barcharts show the proportion of cell-types in each sample (top panel, neutrophils included, bottom panel neutrophils excluded, representing the PBMC population). Each bar is normalised to 1 (100%). Samples are ordered according to disease phenotype (not active TB followed by active TB). The legend lists the cell types identified. Abbreviations: ***neutro***: neutrophils, ***DC act***: activated dendritic cells, ***DC***: non-activated dendritic cells, ***mono***: monocytes, ***NK act***: activated natural killer cells, ***NK***: non-activated natural killer cells, ***B aIGM***: BCR-ligated B cells, ***B***: resting B cells, ***Tc act***: activated CD8 T cell, ***Tc***: resting CD8 T cell, ***Th act***: activated CD4 T cell, ***Th***: resting CD4 T cell

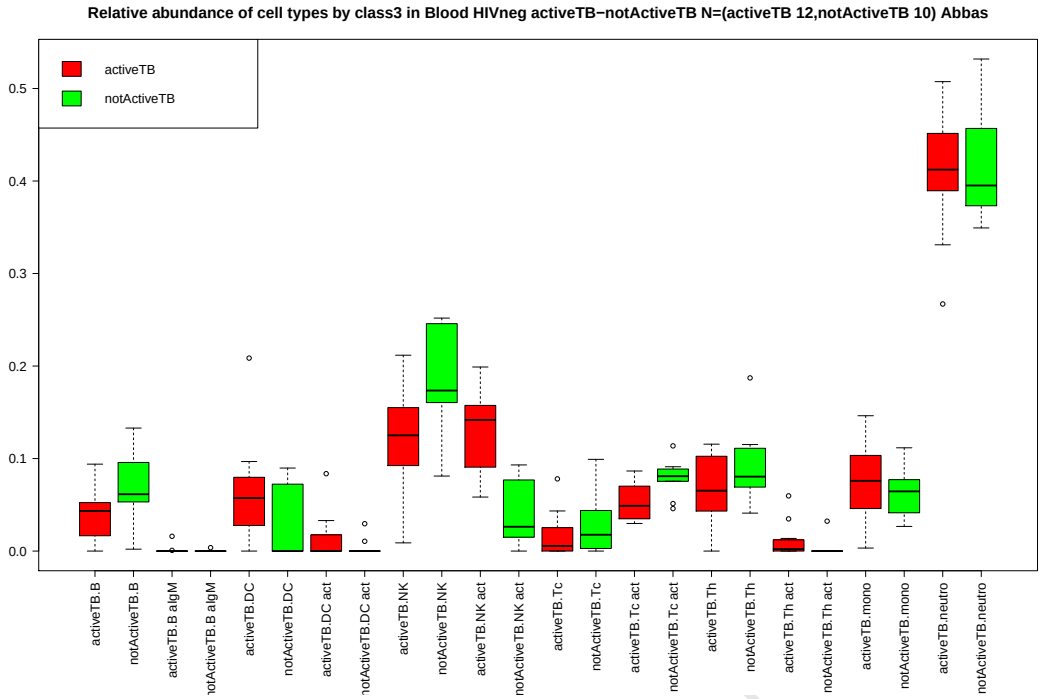
to such pathogens, they require activation by interferons and cytokines derived from macrophages and dendritic cells (e.g. interferon-α, interferon-β (type I interferons), and interleukin-12 (IL-12). Once bound to cells with intracellular pathogens they release cytotoxic granules (containing granzymes and perforin) which kill the infected cell.

In previous studies, NK cells were shown to be activated at the site of disease in a mouse model of tuberculosis, but this did not have an effect on pulmonary bacterial burden [156], questioning the role NK cells might play in protection against active tuberculosis and controlling the disease. An *in vitro* study showed that, in the presence of glutathione and IL-2/ IL-12 stimulation, NK cells inhibit growth of *Mycobacterium tuberculosis* in monocytes; this growth inhibition itself was inhibited when FasL and CD40L, important mediators of NK cell cytotoxicity, were inhibited by antibodies [157]. This leaves open the question whether NK cells are protective in active tuberculosis in humans. In a recent study performed on human tuberculosis site-of-disease samples (lung resection samples) the authors demonstrated that NK cells infiltrated pulmonary granulomatous lesions. In addition, NK cells from healthy volunteers could be stimulated to release IFNγ and TNFα by *M. tuberculosis* H37Rv or *M. bovis* BCG in the presence of IL-2 [158]. The authors also demonstrated variation in responses to TB antigens that correlated with variation in the KIR locus.

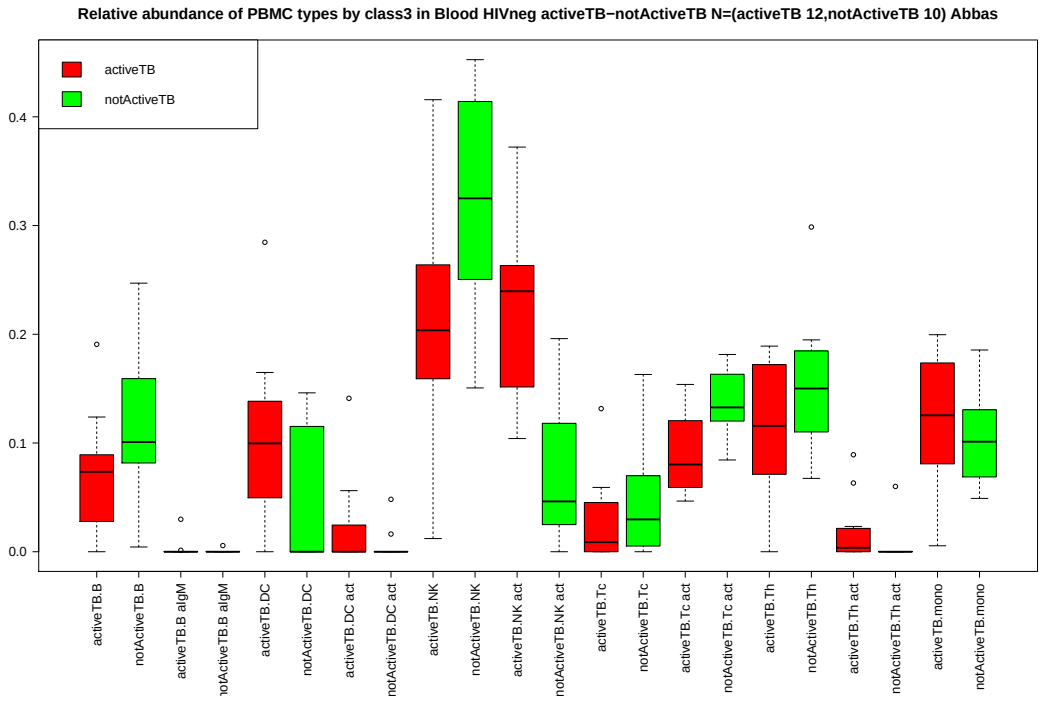
Given the increased proportion of activated NK cells in active tuberculosis suggested by the deconvolution results in light of the studies cited above, the functional consequences of this phenomenon require further exploration.

Table 9.2: **Significance of differences in cell proportions in active TB vs not active TB.** Significant differences are highlighted in yellow.

names	P-value	Bonferroni	Benjamini Hochberg
Th	0.2543	1.0000	0.3987
Th act	0.0728	0.8735	0.1747
Tc	0.2990	1.0000	0.3987
Tc act	0.0044	0.0531	0.0266
B	0.0358	0.4299	0.1075
B aIgM	0.6990	1.0000	0.6990
NK	0.0112	0.1338	0.0446
NK act	0.0004	0.0052	0.0052
mono	0.4176	1.0000	0.5011
DC	0.0943	1.0000	0.1886
DC act	0.2817	1.0000	0.3987
neutro	0.6744	1.0000	0.6990



(a) **Detected types**



(b) **PBMC types**

Figure 9.7: **Boxplots of detected and PBMC types.** The top panel shows box-and-whisker plots for cell proportions by disease phenotype for all 12 detected cell types, and the bottom panel for the PBMC subset. ***Tc act***, ***NK act*** and ***NK*** are statistically significant after multiple testing correction.

9.2.2 Cell-type specific differential expression

Analysis of cell-type specific differential expression was limited to the top 500 differentially expressed genes combined with the probes in the Abbas data that uniquely identifies each cell type. By doing so, sensitivity for detecting subtle changes in gene expression in specific cell types was reduced, but the likelihood of detecting *any* cell-type specific differential expression was increased, as we focus on probes known to be differentially expressed, and remove the majority of probes most likely to contribute to false positive signals caused by excessive noise. So, in the first instance the cell-type specific differential expression output is principally useful to identify cell-types, or groups of cell-types, that are of prime interest for the contrast under investigation.

In several instances, members of the list of identified cell-types were combined into functional classes. This was done where the number of samples in either arm of the contrast was smaller then the number of cell-types. By using a proportion matrix containing fewer classes, even small data subsets were amenable to analysis, assuming that the broader classes used were biologically relevant. Given this caveat, much of the output of cell-type specific differential expression analysis should be regarded as hypothesis-generating at best, and not necessarily a reflection of biological reality.

Figure 9.8 demonstrates some results characteristic for many questions and data subsets. Differential expression of the analysed probes appears to be restricted to mainly NK cells, CD4 T cells and neutrophils, while other lymphocytes, monocytes and dendritic cells do not show evidence in favour of a cell-type specific signal. Given that these latter cell types often become active with regard to tuberculosis at the site of disease, this is not surprising. Deconvolution analysis above showed an increased proportion of activated NK cells, so it is reasonable to hypothesise that the differentially regulated probes for this cell subset are involved in NK cell activation and cytotoxicity. CD4 cells are central to the immune response to tuberculosis. Detection of a differential expression signal in CD4 T cells but not in other lymphocytes may indicate differential recirculation dynamics following exposure to antigen at the site of disease. Figure 9.9 shows the overlap of the top 100 differentially expressed probes for three cell types. 36-46% of probes are unique to these cell-types, highlighting the possibility that this method may be useful for defining specific biological roles in different cell types in tuberculosis using transcriptomic profiles. I will return to

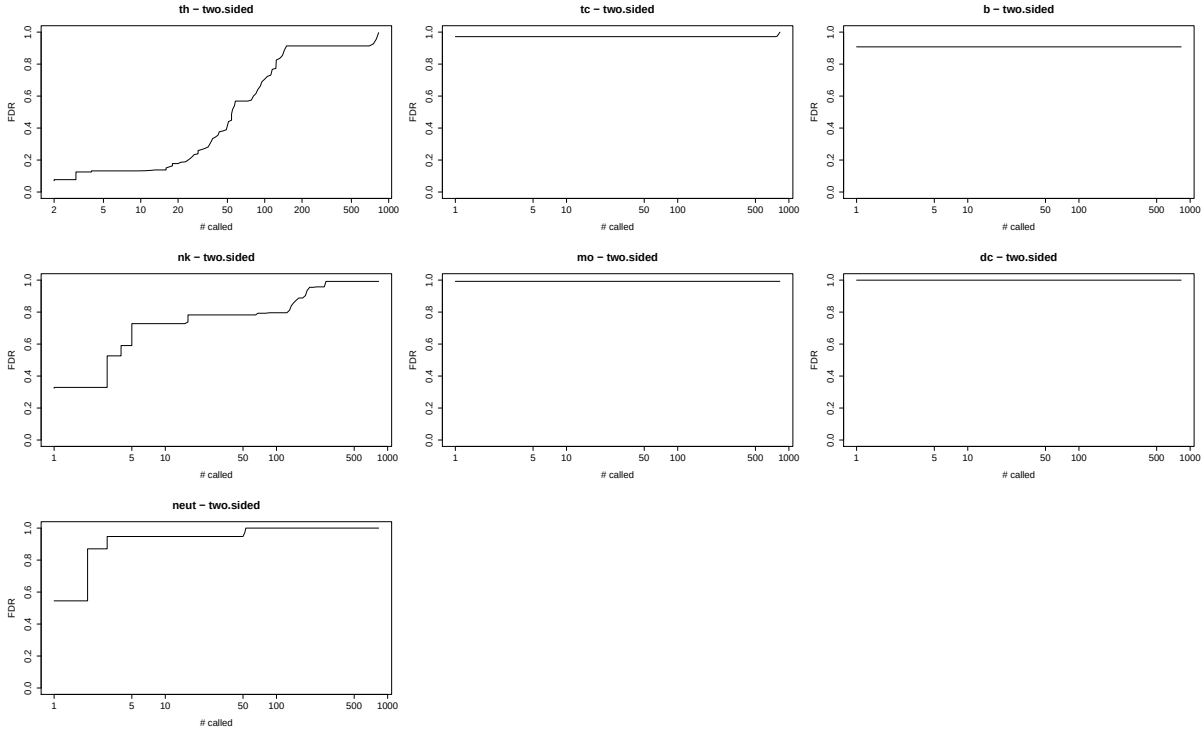


Figure 9.8: **False discovery rate plots after cell-specific DE**

NK cell activation in tuberculosis in the next Chapter.

9.3 Weighted gene co-expression network analysis (WGCNA)

Modular analysis using weighted gene co-expression network analysis was performed next. Using a soft-thresholding power of 5 and an R^2 cutoff of 0.8, sixteen modules were identified. Of the 8000 probes considered for this analysis, 7440 were assigned to one of the modules. The modules[1] vary widely in the number of probes they contain, ranging from 3217 to 24. Figure 9.10 shows the sample dendrogram resulting from hierarchical clustering of the 8000 probes, and below this a heatmap of phenotypic traits.

In Figure 9.11 we clearly see that the contrast variable (class3) is strongly correlated with all modules[2]. This is expected, given that the 8000 probes used for module detection were selected

[1] Modules are arbitrarily named by the software using colour names as labels. As such, the module names are meaningless.

[2] The variable *class3* encodes "activeTB" and "notActiveTB", while the variable *class* encodes "healthy", "LTBI", "PTB" and "TB-PC". The former is used for the contrast in the differential expression anlysis while the latter is used mainly for applying the correct colour to the classes in various figures. Correlations are opposite for these two variables due to the way the categorical variables are converted to numeric values used in the correlation calculations.

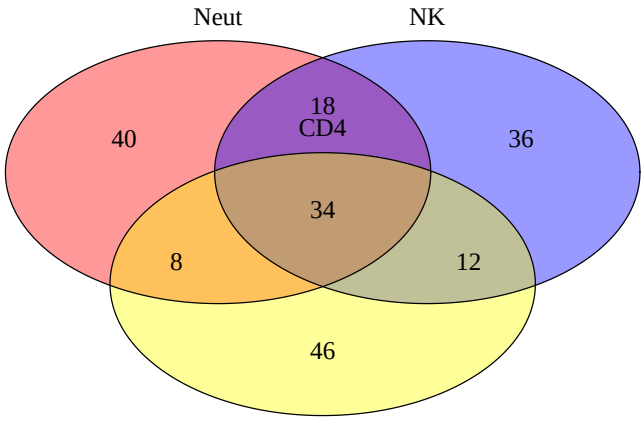


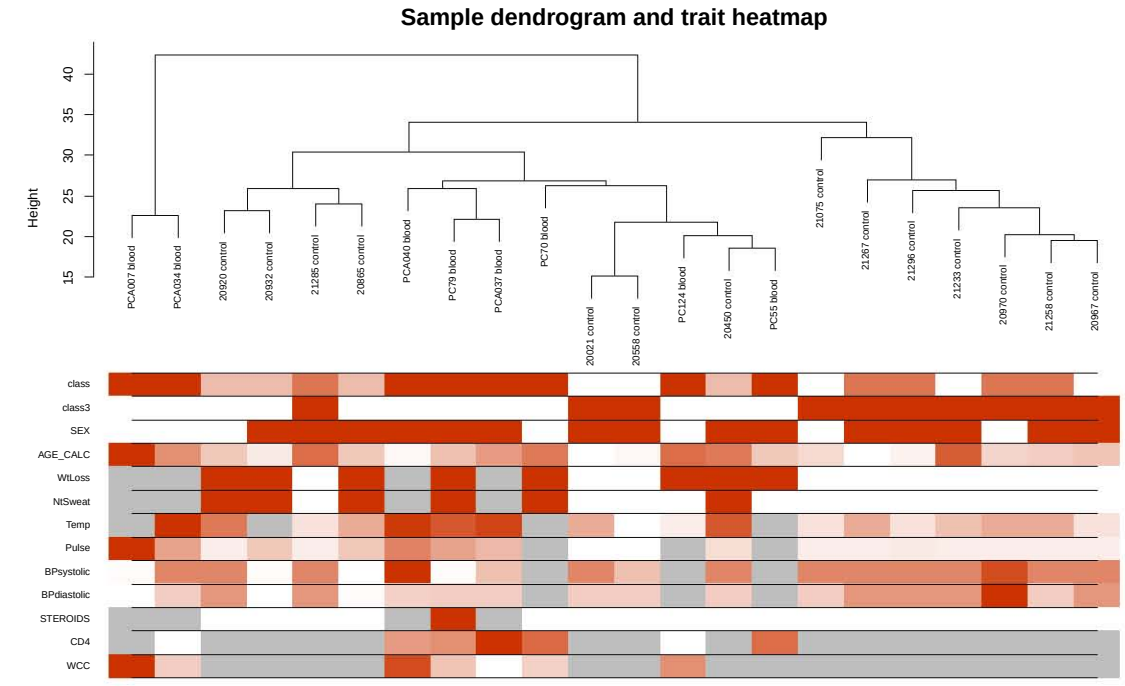
Figure 9.9: **Cell-type specific differential expression.** Venn diagram showing overlap of top 100 DE probes after cell-type specific differential expression analysis for three cell types. Abbreviations: **Neut**: neutrophils; **NK**: natural killer cells; **CD4**: CD4 T cells.

based on their probability of differential expression. Of interest is the fact that some additional variables also show significant correlation with most modules. These variables (like fever, weight loss, night sweats, pulse rate) are expected to correlate with presence and/ or severity of clinical tuberculosis disease. Interestingly, this relationship is not seen with CD4 count, but this may be due to incomplete data as CD4 counts were not measured in many HIV-1 uninfected individuals.

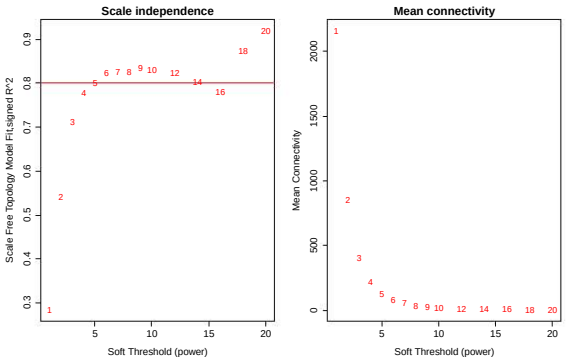
Next, we highlight some features of interesting modules. Associations of individual genes with our trait of interest (clinical tuberculosis) are quantified by the measure "Gene Significance" (GS) (the absolute value of the correlation between the gene and the trait). In addition, a quantitative measure of module membership (MM) is defined as the correlation of the module eigengene and the gene expression profile. For significant modules, the GS and MM measures are correlated; examples for this are shown in Figure 9.12. Highly connected network hub genes tend to have high GS and MM values.

Another way to visualise the modules is to plot the values of the module eigengenes underneath a heatmap of expression data for the particular module, as in Figure 9.13. The barplots suggest that, for these three modules at least, the expression of the module genes is associated with the contrast variable (active TB or not active TB). Boxplots summarise the module eigengene expression data further, and Table 9.3 summarises these comparisons for all detected modules.

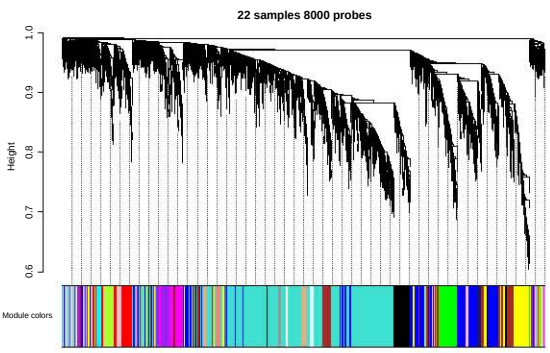
Modules can also be visualised in a network context. Figure 9.14 shows this by the example of the *Red* and *Lightgreen* modules. In both cases, results of differential expression analysis have



(a) **Dendrogram and trait heatmap**



(b) **Scale independence**



(c) **Module detection**

Figure 9.10: **WGCNA: Traits and modules.** In Subfigure a, categorical variables in the phenotype data were converted to numerical variables, and the matrix rescaled to the interval 0 to 1 for each variable. Missing data is shown in grey. Subfigure b shows the data used as basis for selecting the soft-thresholding power used in module detection, and the mean connectivity associated with various soft-thresholding powers. Subfigure c shows the results of module detection by clustering and dynamic branch cutting.

Table 9.3: **Significance of modules for contrast "active TB vs no TB".** Results for comparing expression values of module eigengenes by Kruskal Wallis test. Uncorrected, Bonferroni-corrected and Benjamini-Hochberg-corrected P-values/ Q-values are shown. The *Midnightblue* module is not significant and has been highlighted in yellow. All module names are simply arbitrary colour labels.

	Module name	P-value	Bonferroni	Q-value
1	green	0.0090	0.1971	0.0104
2	turquoise	0.0003	0.0066	0.0007
3	black	$5.8765 \cdot 10^{-05}$	0.0013	0.0004
4	lightyellow	0.0034	0.0756	0.0047
5	cyan	$9.2786 \cdot 10^{-05}$	0.0020	0.0004
6	red	$9.2786 \cdot 10^{-05}$	0.0020	0.0004
7	blue	$9.2786 \cdot 10^{-05}$	0.0020	0.0004
8	midnightblue	0.0591	1.0000	0.0591
9	darkred	0.0112	0.2454	0.0123
10	tan	0.0006	0.0132	0.0010
11	pink	0.0006	0.0132	0.0010
12	brown	0.0011	0.0248	0.0018
13	purple	0.0001	0.0031	0.0005
14	magenta	0.0003	0.0066	0.0007
15	yellow	0.0090	0.1971	0.0104
16	greenyellow	0.0026	0.0582	0.0039
17	grey60	0.0056	0.1242	0.0073
18	lightcyan	0.0169	0.3718	0.0177
19	lightgreen	0.0006	0.0132	0.0010
20	salmon	0.0003	0.0066	0.0007
21	royalblue	0.0004	0.0095	0.0009
22	grey	$9.2786 \cdot 10^{-05}$	0.0020	0.0004

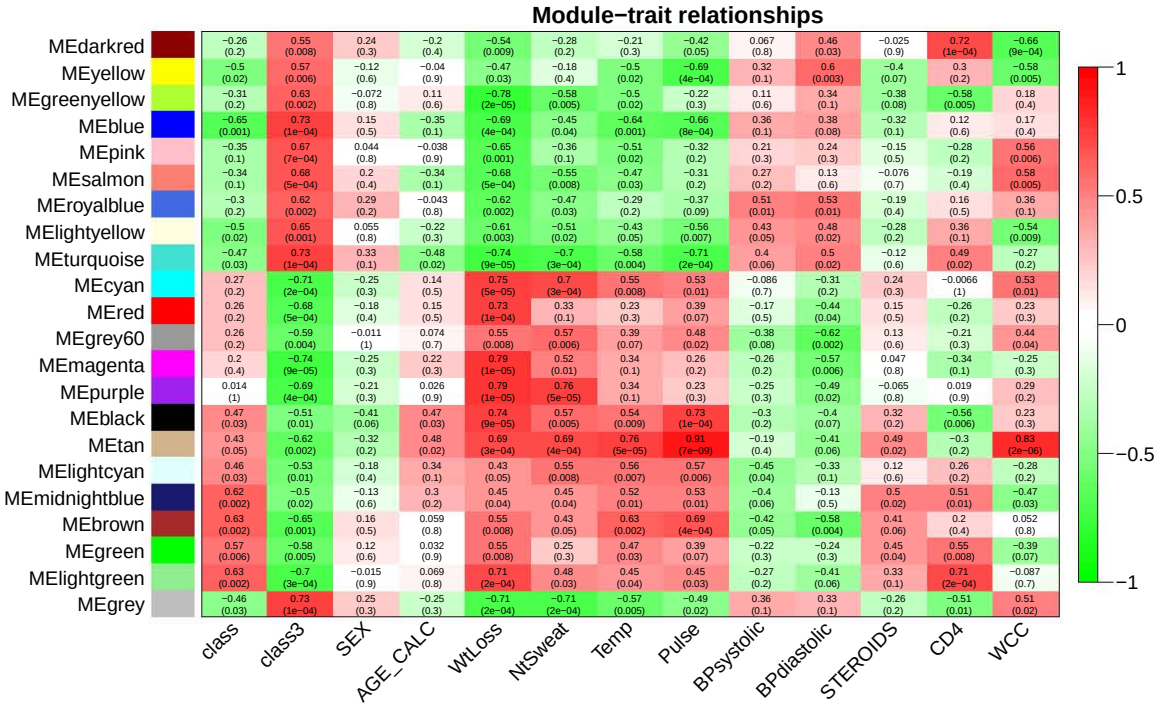
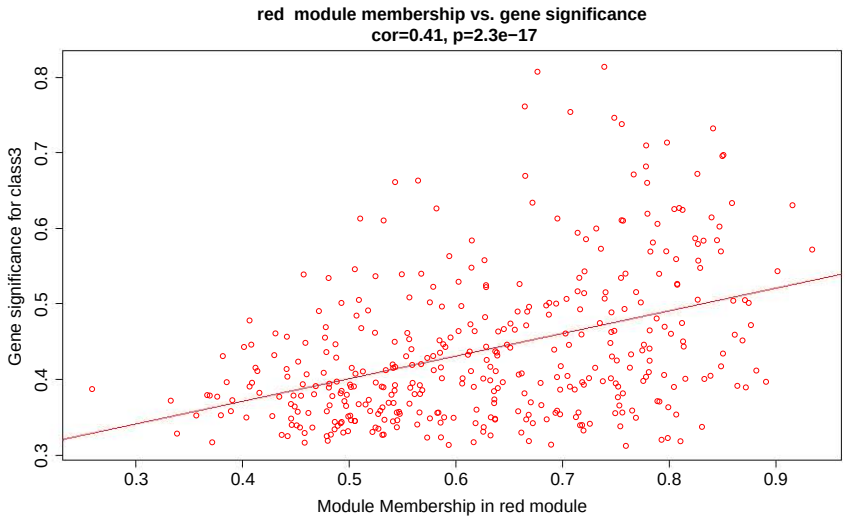


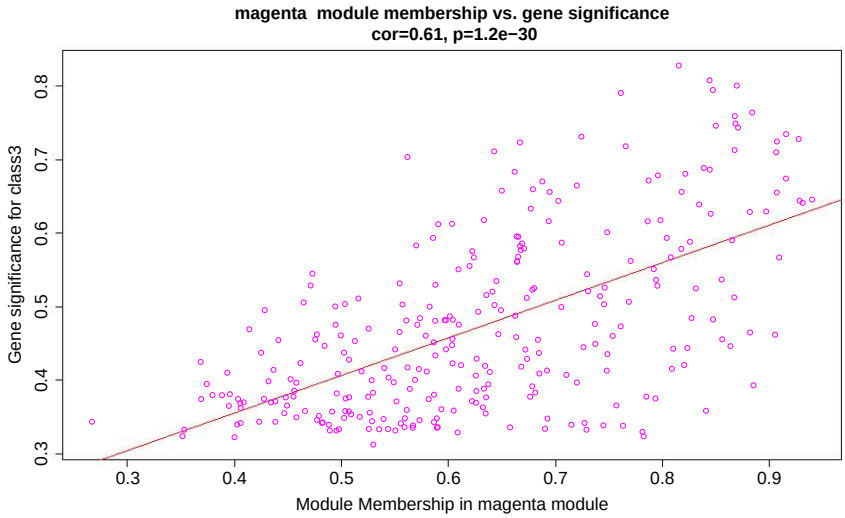
Figure 9.11: **Module/ trait relationships.** This heatmap shows the Pearson correlation coefficients (R) and their P-values for a number of phenotypic variables as correlated with the detected modules. Colours: green = positive correlation; red = negative correlation.

been mapped onto the nodes.

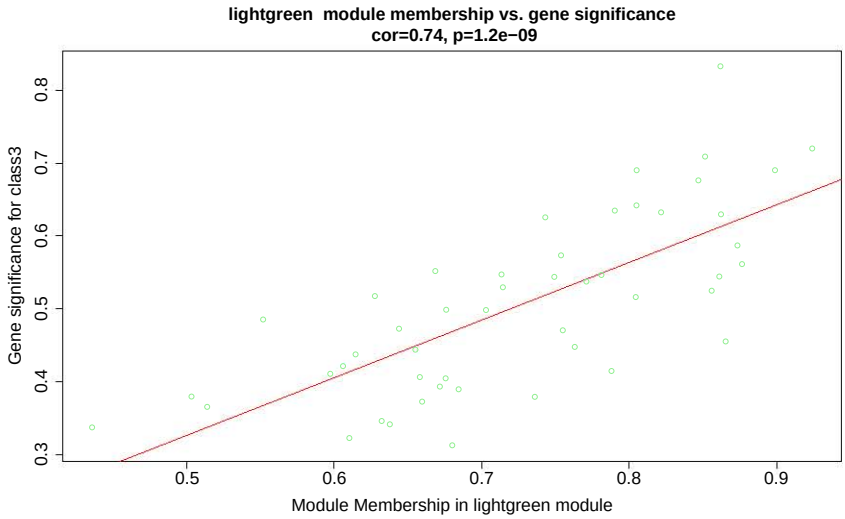
The results of WGCNA-analysis demonstrate that modules significant in tuberculosis can be identified. These modules await further characterisation at a more detailed level by identifying the genes central to the modules' functioning, as well as any pathways or groups of pathways associated with the modules.



(a) **Red module**

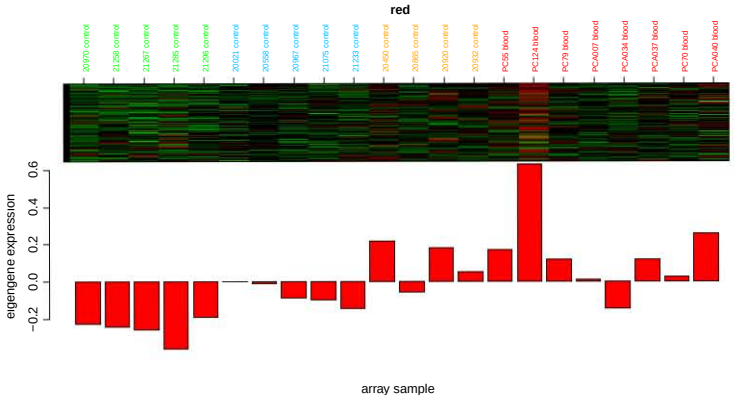


(b) **Magenta module**

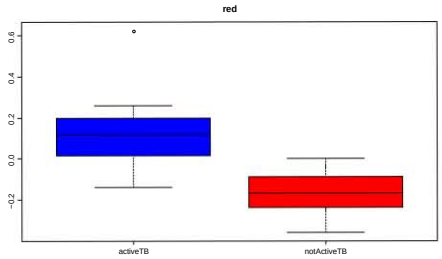


(c) **Lightgreen module**

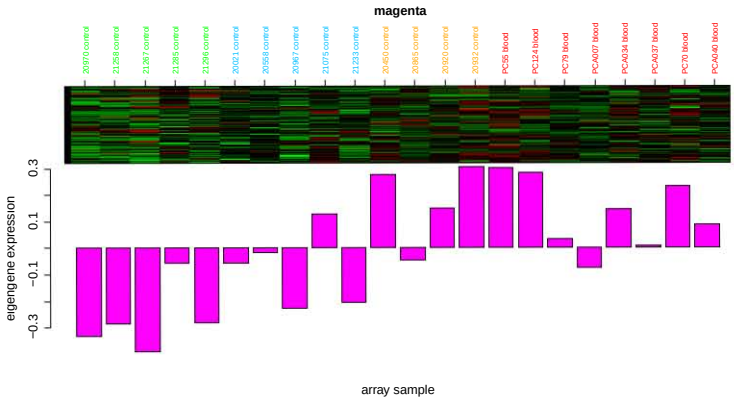
Figure 9.12: **Top 3 GS/MM plots** This figure demonstrates correlation of the measures *Gene Significance vs. Module Membership* (GS/MM). The results for three modules are shown.



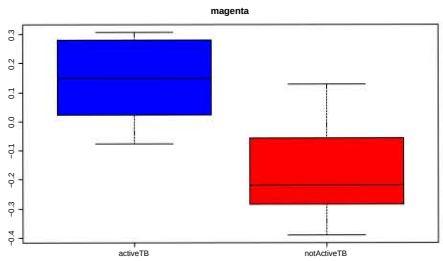
(a) **Red module: eigengene expression**



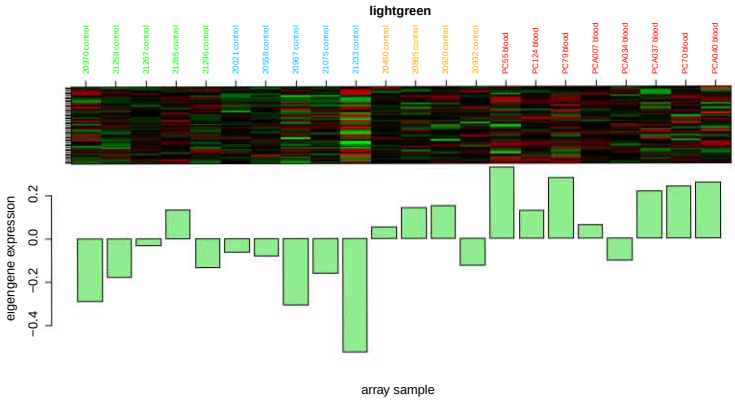
(b) **Eigengene boxplot**



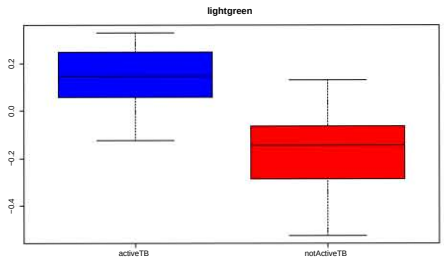
(c) **Magenta module: eigengene expression**



(d) **Eigengene boxplot**

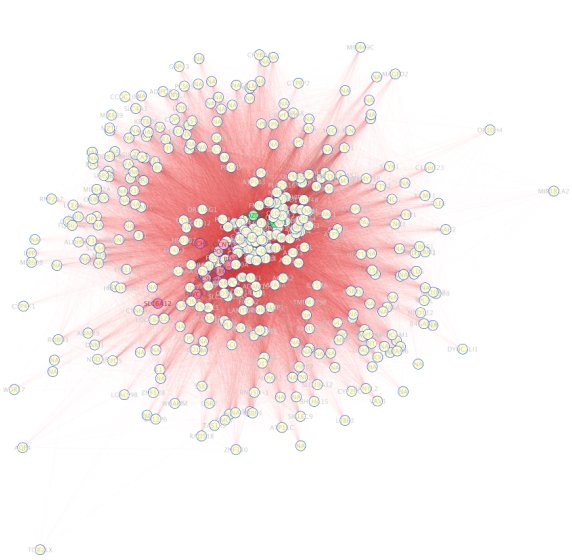


(e) **Lightgreen module: eigengene expression**

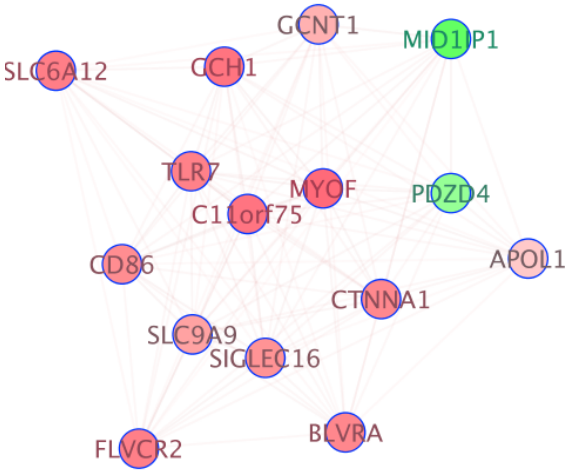


(f) **Eigengene boxplot**

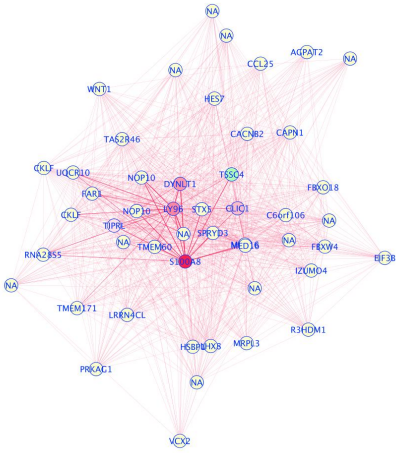
Figure 9.13: **Three significant modules.** Results for three modules are demonstrated in more detail. Bar plots of eigengene expression combined with a heatmap of the probes making up the module are shown on the left. The right panel shows a box and whisker chart of a summary of all eigengenes (one per sample) in each of the two contrast classes. Test statistics for the median difference are shown in Table 9.3. The sample names are coloured by TB class (*green* = healthy, *blue* = LTBI, *orange* = PTB, *red* = TB-PC). Note that for these three modules their respective eigengenes are downregulated in *not active TB* and upregulated in *active TB*



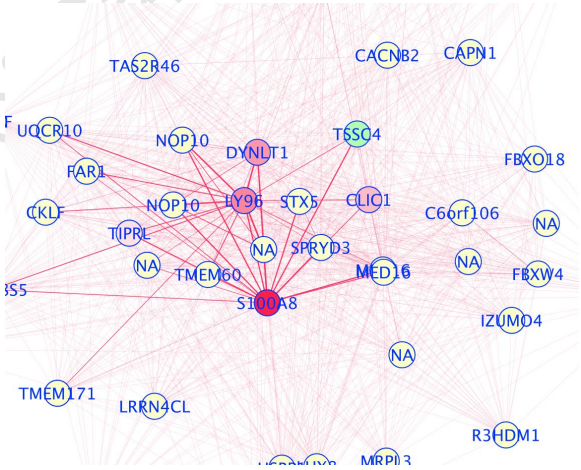
(a) **Red module, complete network**



(b) **Red module, zoomed in and only showing genes that were differentially expressed**



(c) **Lightgreen module, complete network**



(d) **Lightgreen module, zoomed in view. Differentially expressed genes are highlighted. These genes appear to be hubs of the network.**

Figure 9.14: **Weighted gene co-expression networks.** Networks of gene co-expression were generated in Cytoscape. Nodes reported as differentially expressed in earlier analysis are coloured on a green (downregulated in active TB) to red (upregulated in active TB) colour scale. Network edges are weighted by the strength of association between any of their nodes.

Part IV

IMPI-MA: Results in breadth

10 Contrasts involving tuberculosis

In this and the following two chapters I present a broad impression of results for all questions that form part of the analysis. The results are mainly presented in tabular form and summary images. The emphasis here is on comparison of results (rendered in broad brushstrokes as opposed to the level of detail used in the preceding chapter); in particular, the results for two datasets in each question will be juxtaposed, in order to highlight similarities and differences.

This Chapter deals with questions relating to active and latent tuberculosis, effects of pulmonary vs. extrapulmonary tuberculosis and differences between haemodynamic phenotypes of TB-PC.

General comment: Due to heterogeneity of the IMPI-MA data, many analyses lack adequate statistical power. Some data subsets lend themselves to splitting into "training" and "test" sets, as sufficient numbers of samples are available. Future work will address this by splitting IMPI-MA data subsets into training and test sets or applying the signatures to other extant microarray datasets (Berry et al, Maertzdorf et al). The current work does not provide for validation of signatures, and for this reason should be considered as an exploration of the "space" of signatures in HIV-TB.

10.1 Question 1: Tuberculosis

In this Section I show the results for the contrast "active TB vs. not-active TB", in order to address questions regarding the biology of active tuberculosis disease. Related to this question are differences in biology that are due to the site of active disease (lung vs. pericardium), biological features of LTBI, and differences in cellular and transcriptional responses in two different haemodynamic presentations of TB-PC. These three additional questions will be addressed in the following sections.

10.1.1 Included patients and samples

I examined the contrast active vs not active TB in blood in two contexts: HIV-1 uninfected and HIV-1 co-infected individuals. Table 10.1 show the baseline demographic and clinical characteristics of the study participants on whose samples the microarray analysis is based. This table has been structured to facilitate comparisons between the two sample sets as well as for the contrast variable within each sample set[1]. Three variables were examined: age, sex and CD4 count.

In comparisons across the two study groups, HIV-1/*M. tuberculosis* co-infected individuals were younger than their HIV-1 uninfected counterparts, but this difference failed to reach statistical significance. HIV-1 infected individuals were predominantly female and HIV-1/*M. tuberculosis* co-infected individuals had lower CD4 counts than their HIV-1 uninfected counterparts. Comparing these three variables within each study group, the HIV-1 uninfected group was well matched for age and sex (CD4 count could not be assessed), but the HIV-1 infected group was only matched for age, as all cases of not active TB were female, and the active TB group had significantly lower median CD4 counts. The sex bias in the HIV-1 infected not active TB group is explained by patterns of health-seeking behaviour in asymptomatic HIV-1 infected persons; in South Africa men infrequently attend wellness clinics if asymptomatic, therefore females are much more readily recruited into this category. Future studies should employ stratified sampling in order to avoid this type of bias. In summary, when examining the contrast *activeTB vs not active TB* we should be aware of possible bias introduced by different CD4 counts and male to female ratios in the data in the HIV-1 infected group, and when comparing the results of the two analyses, we should consider the possibility of sex and CD4 count contributing to any differences in results.

10.1.2 Overall results

In Table 10.2 I show the comparison of the main results of the analytic pipeline. A much higher number of probes (5404 vs 369) were differentially expressed in the HIV-1 infected group at a false discovery rate cutoff of 0.05. This is likely due to several factors. Firstly, the second analysis had greater statistical power to detect differences, as 3.3 times the number of study participants in the HIV-1 uninfected group were available. Secondly, as noted above, the HIV-1 infected study

[1] While the table is hard to read, it summarises a lot of data, and allows the comparison of variables within and between datasets. Comparisons between datasets are read horizontally, and comparisons within datasets are read vertically

Table 10.1: **Clinical characteristics: active TB vs not active TB.** Two-way comparison of demographic variables and CD4 count between and with groups of study participants. P-values less than 0.05 are regarded as significant.

	Blood HIVneg activeTB-notActiveTB			Blood HIVpos activeTB-notActiveTB			Pval	Test
Age	Median	LQ	UQ	Median	LQ	UQ	Pval	Test
age_All	33.85	26.55	51.48	32.98	27.2	38.18	0.422	Kruskal-Wallis rank
age_ activ-eTB	40.67	33.69	52.86	33.3	29.6	38.6	0.065	Kruskal-Wallis rank
age_ not-ActiveTB	29.9	22.98	34.2	29.5	25.75	37.3	0.584	Kruskal-Wallis rank
age_P value	0.086			0.084				
age_Test	Kruskal-Wallis rank			Kruskal-Wallis rank				
Sex	ratio F/M	Female	Male	ratio F/M	Female	Male	Pval	test
sex_All	0.467	7	15	2.650	53	20	0.001	2-sample test for eq
sex_ activ-eTB	0.714	5	7	1.300	26	20	0.553	2-sample test for eq
sex_ not-ActiveTB	0.250	2	8	Inf	27	0	0.000	2-sample test for eq
sex_P value	0.531			0.000				
sex_Test	2-sample test for eq			2-sample test for eq				
CD4	Median	LQ	UQ	Median	LQ	UQ	Pval	Test
CD4_All	566	404.5	653	260	117	361.5	0.007	Kruskal-Wallis rank
CD4_ activeTB	566	404.5	653	167	86	291.5	0.001	Kruskal-Wallis rank
CD4_ not-ActiveTB				333	307.5	467	NA	NA
CD4_P value	NA			0.000				
CD4_Test	NA			Kruskal-Wallis rank				

participants grouped by contrast variable were unmatched for sex and CD4 count, and finally, it is plausible that HIV-1 co-infection introduces additional variability to gene expression. Following deconvolution analysis, proportions for 3 of 12 cell types were significantly different in HIV-1 uninfected individuals, whereas 8 of 14 cell types differed in proportion in HIV-1 infected individuals. For the WGCNA analysis the same soft thresholding power was selected; a similar number of modules was detected in both cases, with at least 93% of probes assigned to modules. The following subsection examines some results in more detail.

Table 10.2: **Overall results: active TB vs not active TB.** Output of the analysis script, showing the numbers of included samples, and statistics for differential expression, deconvolution and gene co-expression network analysis. P-values less than 0.05 are regarded as significant.

Question 1: Tuberculosis		
Contrast: Active TB vs. not Active TB		
Dataset	1	2
Context	Blood, HIV neg	Blood, HIV pos
N: Active TB	12	46
N: not Active TB	10	27
Analysis 1: Differential expression		
N significant probes (unadjusted)	4121	8847
N significant probes (BH adjusted)	369	5405
N probes used (heatmaps, csDE)	369	2000
Analysis 2: Deconvolution		
Deconvolution successful	yes	yes
N of N significant for contrast (BH)	3 of 12	8 of 14
N of N PBMC significant for contrast (BH)	2 of 11	5 of 13
Cell-specific DE successful	yes	yes
Number of cell types	7	7
N cell types that reach FDR < 0.4	2	3
Analysis 3: WGCNA		
R^2 cutoff for scale-free approximation	0.8	0.8
Soft threshold power	5	5
Number of modules	21	16
Probes assigned to modules (N, %)	7937 (99)	7440 (93)
GO: Entrez IDs submitted	4382	4730
GO: Entrez IDs mapped	2920	3182
N (%) modules cor GS/MM sig	17 (81)	12 (75)
N (%) modules sig for contrast (BH)	20 (95)	16 (100)

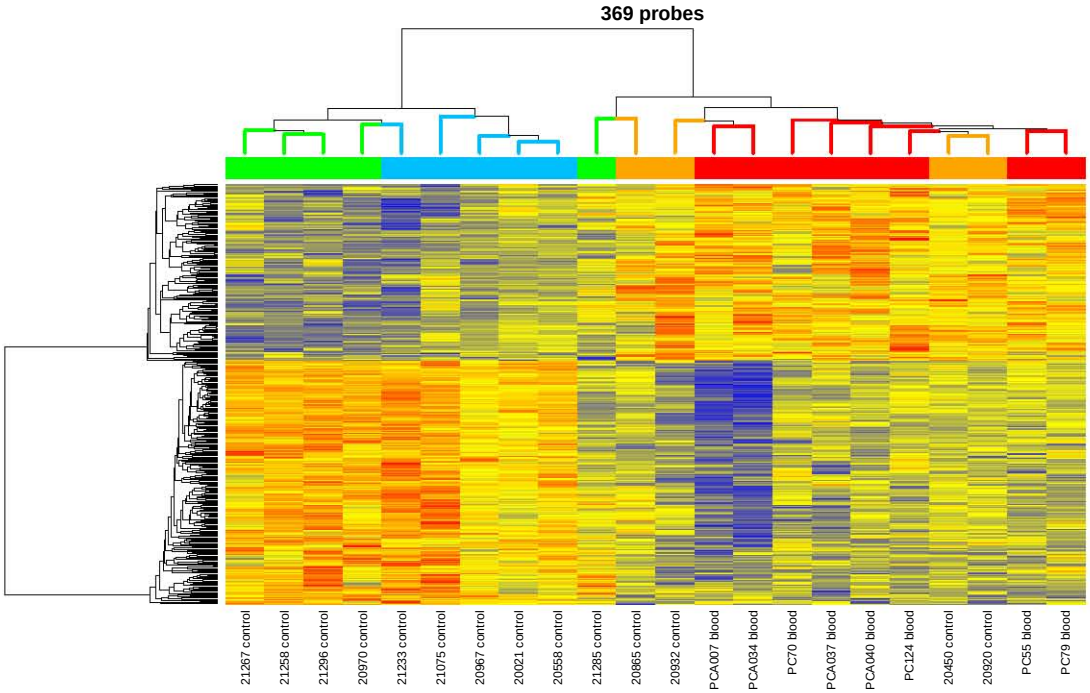
10.1.3 Results for context: HIV (datasets: Blood, HIV negative and Blood, HIV positive)

How well do these differentially expressed probes classify the input data? Figure 10.1 shows heatmaps of the significantly differentially expressed probes for the HIV-1 uninfected group, and the top 2000 differentially expressed probes for the HIV-1 infected group. In the first case the input data is classified well (one "healthy" sample is misclassified), and in the second case, two distinct clusters are seen: one cluster only contains active TB cases, and the second six active TB cases in addition to all the not-active TB cases. The reason for this misclassification may lie in the fact that either an insufficient number of probes were selected, or that the signal due to HIV-1 infection interferes with the active TB signal.

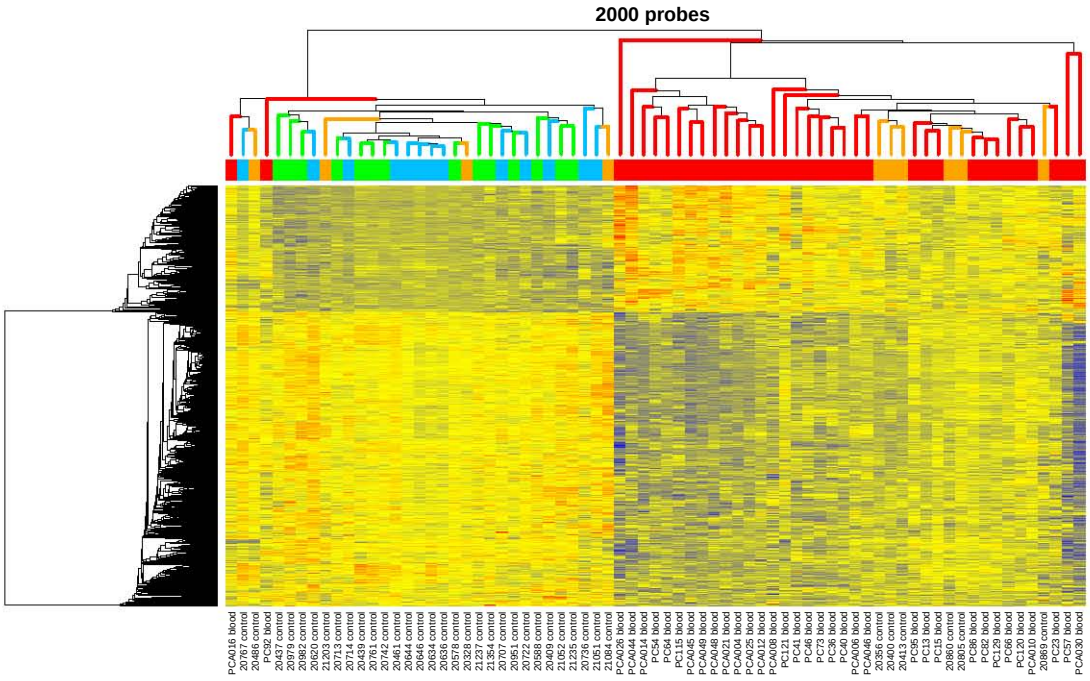
False discovery rate (FDR) plots for the two analyses (Figure 10.2) offer an additional perspective on the results. While the signal for cell-specific differential expression is seen in CD4 T cells, NK cells and to a lesser extent neutrophils in HIV-1 uninfected individuals (see also Chapter 9), we find stronger signals in CD8 T cells, neutrophils, monocytes and dendritic cells in HIV-1 infected individuals, while the signal for differential expression in NK cells is much less convincing. This may be due to effects of HIV-1 on gene expression in these cell subsets in the context of a TH1-predominant inflammatory background. It is well described that *Mycobacterium tuberculosis* and HIV-1 interact significantly (See Chapter 1). Both monocytes and dendritic cells may become infected with HIV-1 [159], and it is possible that the differential expression in these cell types is due to effects of HIV-1 which are only seen in the context of active tuberculosis. A possible effect of HIV-1 on monocytes and dendritic cells is the induction of apoptosis [160], and detection of induction of pro-apototic factors in these cell subsets would provide additional evidence for this.

Module detection was similar in both groups as see in Figure 10.3. The functional characteristics of the identified modules are still to be determined.

Finally I looked at the overlap of differentially expressed probes in the two contexts. 237 probes were found to overlap between the 369 probes found in HIV-1 uninfected individuals and the top 2000 probes in HIV-1 infected individuals, accounting for 64% of the probes in the first set. The non-overlapping probes may represent noise or HIV-1 specific effects, while the overlapping probes represent effects driven by active tuberculosis regardless of HIV-1 infection.

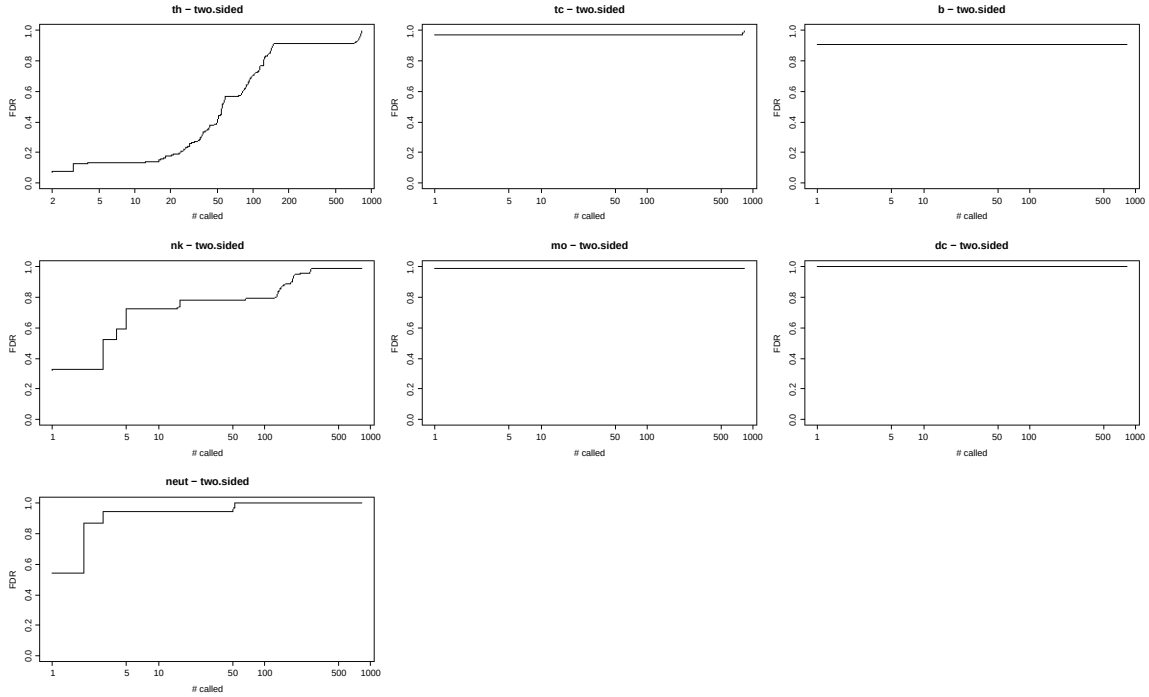


(a) **Heatmap of differentially expressed probes, HIV-1 uninfected**

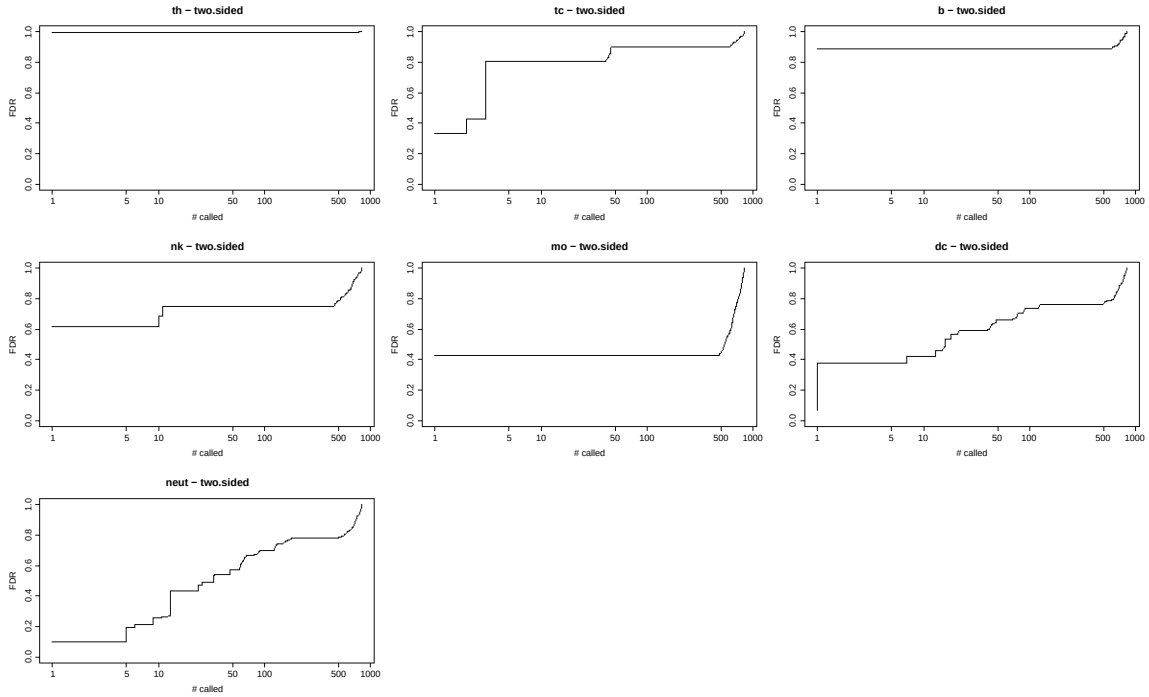


(b) **Heatmap of differentially expressed probes, HIV-1 infected**

Figure 10.1: **Results for contrast *TB status* in contexts *HIV negative* and *HIV positive*.** The heatmaps show good separation of samples regardless of HIV status. The HIV negative group was discussed in Chapter 9. In comparison, the HIV positive group had 14.6 times the number of differentially regulated probes; only the top 2000 probes are shown here. Values are row-scaled, and range from low (blue) to intermediate (yellow) to high (red). Samples are clustered using Spearman rank correlation, and probes are clustered using Pearson correlation. Sample colours: green: *healthy*, blue: *LTBI*, orange: *PTB*, red: *TB-PC*. Plot titles reflect the number of included probes.

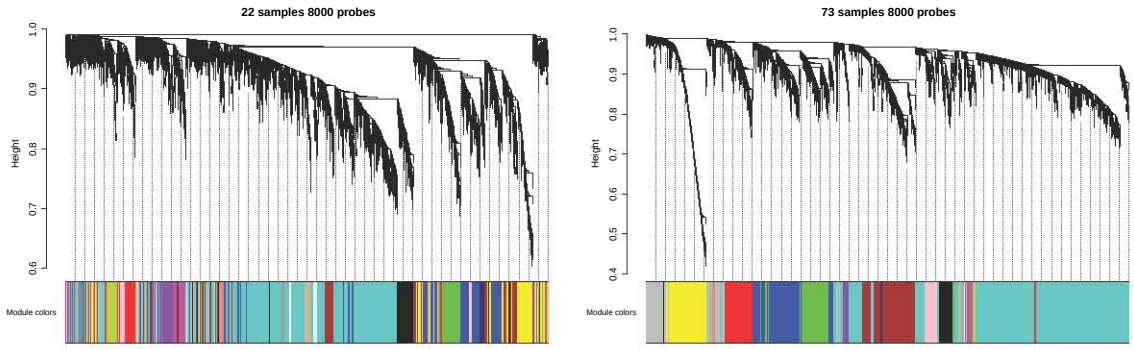


(a) **False discovery rate plot HIV-1 uninfected**



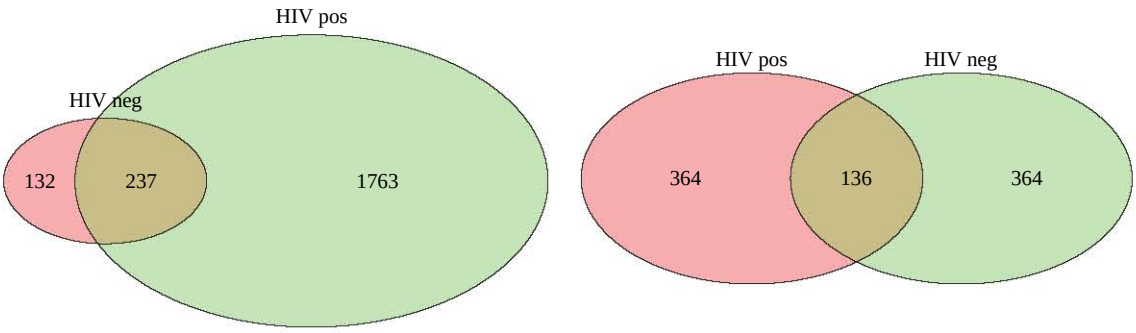
(b) **False discovery rate plot HIV-1 infected**

Figure 10.2: **Cell-type specific differential expression.** False discovery rate plots for the two analyses. For each plot the x-axis shows the number of differentially expressed probes called, given the false discovery rate shown on the y-axis.



(a) Module clustering HIV neg (b) Module clustering HIV pos

Figure 10.3: **Module clustering.** The plots show the dendrograms resulting form hierarchical clustering of the 8000 probes used in the analysis. Colours indicate the modules to which probes were assigned; these colours correspond to the module names.



(a) **369 significantly differentially expressed probes** (b) **Top 500 differentially expressed probes**

Figure 10.4: **Overlap of differentially expressed probes.**

10.2 Question 2: PTB and extrapulmonary TB

In this Section I examine the contrast "PTB vs TB-PC". This looks for differences in transcriptional response detectable in blood based on the site of MTB infection. The underlying hypothesis for this contrast is that the site of disease elicits an additional signal in the transcriptomic data over and above that of TB itself.

10.2.1 Included patients and samples

Table 10.3 summarises the variables age, sex and CD4 count across two groups of study subjects (HIV-1 uninfected and -infected) as well as within each of the two groups across the contrast variable *PTB vs TB-PC*. HIV-1 infected individuals presenting with TB-PC were younger than their HIV-1 uninfected counterparts, and HIV-1 infected individuals presenting with PTB were all female while the female to male ratio in the HIV-1 uninfected group was 0.33. As expected, median CD4 counts in HIV-1 infected individuals presenting with TB-PC were significantly lower than in HIV-1 uninfected cases. The HIV-1 infected group was matched for age, but not sex or CD4 count while HIV-1 uninfected group was matched for age and sex (matching for CD4 count could not be assessed due to missing data). The finding that HIV-1 infected individuals presenting with TB-PC, a form of extrapulmonary TB, had lower CD4 counts than those presenting with pulmonary disease is in line with expectation. In summary, differences in CD4 count and sex should be considered as potential biasing factors in this analysis.

10.2.2 Overall results

Table 10.4 lists the overall statistics of running the analytic pipeline on the two datasets. It should be noted that the sample sizes in the HIV-1 uninfected group are very small, and as such, any results should be regarded as hypothesis generating at best. Another potentially important biasing factor is the fact that the RNA from the two groups was collected at different times in different study contexts, and was extracted at different time points. Of all analyses, this one is probably most susceptible to technical bias.

The number of differentially expressed probes found in the two datasets after multiple testing correction was 38 (HIV-1 infected) and 0 (HIV-1 uninfected), respectively. In order to explore

Table 10.3: **Clinical characteristics: PTB vs TB-PC.** Two-way comparison of demographic variables and CD4 count between and with groups of study participants. P-values less than 0.05 are regarded as significant.

	Blood HIVpos PTB-TBPC			Blood HIVneg PTB-TBPC			Pval	Test
Age	Median	LQ	UQ	Median	LQ	UQ	Pval	Test
age_All	33.3	29.6	38.6	40.67	33.69	52.86	0.065	Kruskal-Wallis rank
age_ CON_PTB	34.25	28.12	42.27	33.85	31.78	38.6	0.888	Kruskal-Wallis rank
age_ TBPC	33.13	29.7	38.32	46.6	35.3	54.01	0.036	Kruskal-Wallis rank
age_P value	0.749			0.308				
age_Test	Kruskal-Wallis rank			Kruskal-Wallis rank				
Sex	ratio F/M	Female	Male	ratio F/M	Female	Male	Pval	test
sex_All	1.300	26	20	0.714	5	7	0.553	2-sample test for eq
sex_ CON_PTB	Inf	10	0	0.333	1	3	0.018	2-sample test for eq
sex_ TBPC	0.800	16	20	1.000	4	4	1.000	2-sample test for eq
sex_P value	0.006			0.836				
sex_Test	2-sample test for eq			2-sample test for eq				
CD4	Median	LQ	UQ	Median	LQ	UQ	Pval	Test
CD4_All	167	86	291.5	566	404.5	653	0.001	Kruskal-Wallis rank
CD4_ CON_PTB	322	267.5	454.5				NA	NA
CD4_ TBPC	110	78	237.2	566	404.5	653	0.000	Kruskal-Wallis rank
CD4_P value	0.000			NA				
CD4_Test	Kruskal-Wallis rank			NA				

classification of the input data and probe overlap, I selected the top 300 differentially expressed probes (using as ranking measure the log-odds of differential expression), bearing in mind that many of these probes are potentially (but not definitely) false positives. The sample size in the HIV-1 uninfected group was too small for meaningful cell-specific differential expression; this will not be reported here. Finally, a high number of modules was identified in the HIV-1 uninfected group (67), with all but two probes assigned to modules. In contrast, only 62.1% of probes were assigned to modules in the HIV-1 infected group.

Table 10.4: **Overall results: PTB vs TB-PC.** Output of the analysis script, showing the numbers of included samples, and statistics for differential expression, deconvolution and gene co-expression network analysis. P-values less than 0.05 are regarded as significant.

Question 2: TB Site		
Contrast: PTB vs. TB-PC		
Dataset	1	2
Context	Blood, HIV pos	Blood, HIV neg
N: PTB	10	4
N: TB-PC	36	8
Analysis 1: Differential expression		
N significant probes (unadjusted)	3593	2260
N significant probes (BH adjusted)	38	0
N probes used (heatmaps, csDE)	300	300
Analysis 2: Deconvolution		
Deconvolution successful	yes	yes
N of N significant for contrast (BH)	3 of 13	0 of 12
N of N PBMC significant for contrast (BH)	2 of 12	0 of 11
Cell-specific DE successful	yes	no
Number of cell types	7	NA
N cell types that reach FDR < 0.4	1	NA
Analysis 3: WGCNA		
R^2 cutoff for scale-free approximation	0.8	0.8
Soft threshold power	7	9
Number of modules	18	67
Probes assigned to modules (N, %)	4971 (62.1)	7998 (99.99)
GO: Entrez IDs submitted	4406	4048
GO: Entrez IDs mapped	2925	2662
N (%) modules cor GS/MM sig	11 (61)	44 (66)
N (%) modules sig for contrast (BH)	18 (100)	37 (55)

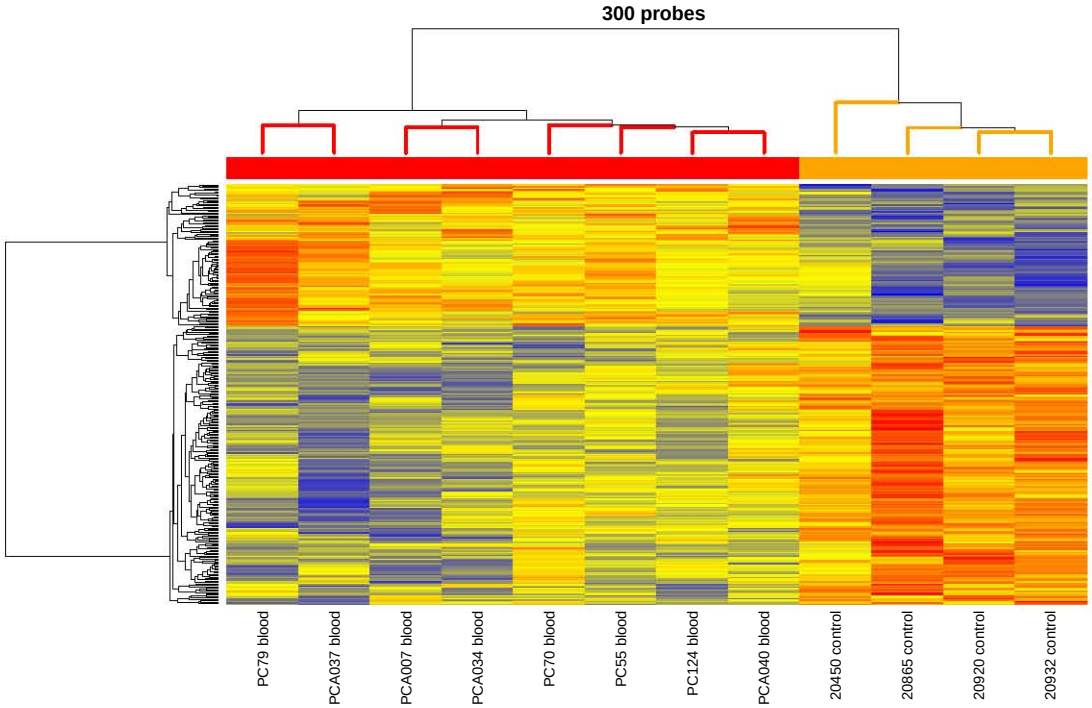
10.2.3 Results for context HIV: (datasets: Blood, HIV negative and Blood, HIV positive)

The heatmaps in Figure 10.5 show that in both cases, the top 300 selected probes do quite well in classifying the input data. There are no misclassifications in the HIV-1 uninfected group, and two cases of PTB are misclassified as TB-PC in the HIV-1 infected group. Given the higher risk for extrapulmonary tuberculosis in HIV-1 co-infection, concurrent pulmonary and extrapulmonary tuberculosis in these cases can not be excluded as an explanation for this. In any event, any difference in transcriptional profiles that are due to the site of TB disease is likely to be small, with a much more subtle signal than active TB vs. not active TB.

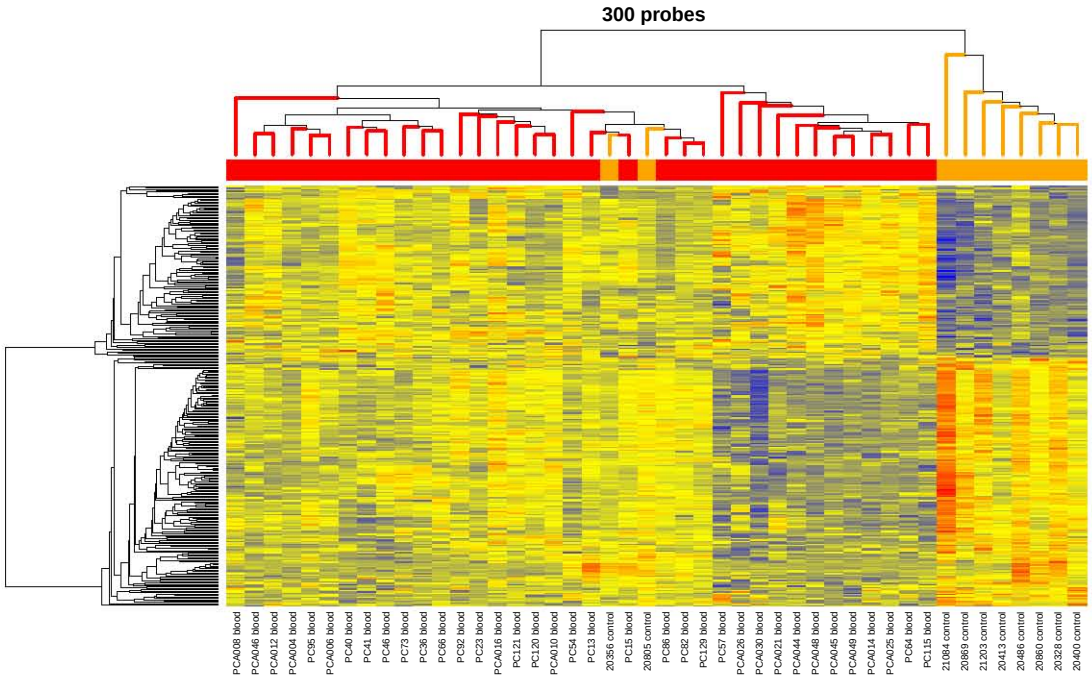
Cell-type specific differential expression analysis offers additional clues about the significance of any detectable differential expression between sites of tuberculosis disease (Figure 10.6). Due to small sample numbers, no estimate could be made in the HIV-1 uninfected group, but in HIV-1 infected individuals the cell-type specific signal is strongest in CD8 T cells. Importantly, given that the TB-PC cases had lower CD4 counts than the PTB cases, the signal may in fact be due to a significant effect of HIV, rather than a signal due to tuberculosis disease. With this interpretation, extrapulmonary tuberculosis would serve as a marker of more advanced HIV disease. More likely, the signal is of mixed origin, with differential severity of HIV-1 infection, *M. tuberculosis* organism burden and location-specific changes.

Figure10.7 highlights differences in module detection in the two datasets. A much large number of modules is found in HIV-1 uninfected individuals.

Figure 10.8 is informative regarding the biological significance of the two signatures; of the top 300 differentially expressed probes, only 14 probes overlap. While it is likely that the selected probes are biologically informative, the added effect of HIV-1 infection in the context of extrapulmonary tuberculosis probably leads to the lack of overlap.

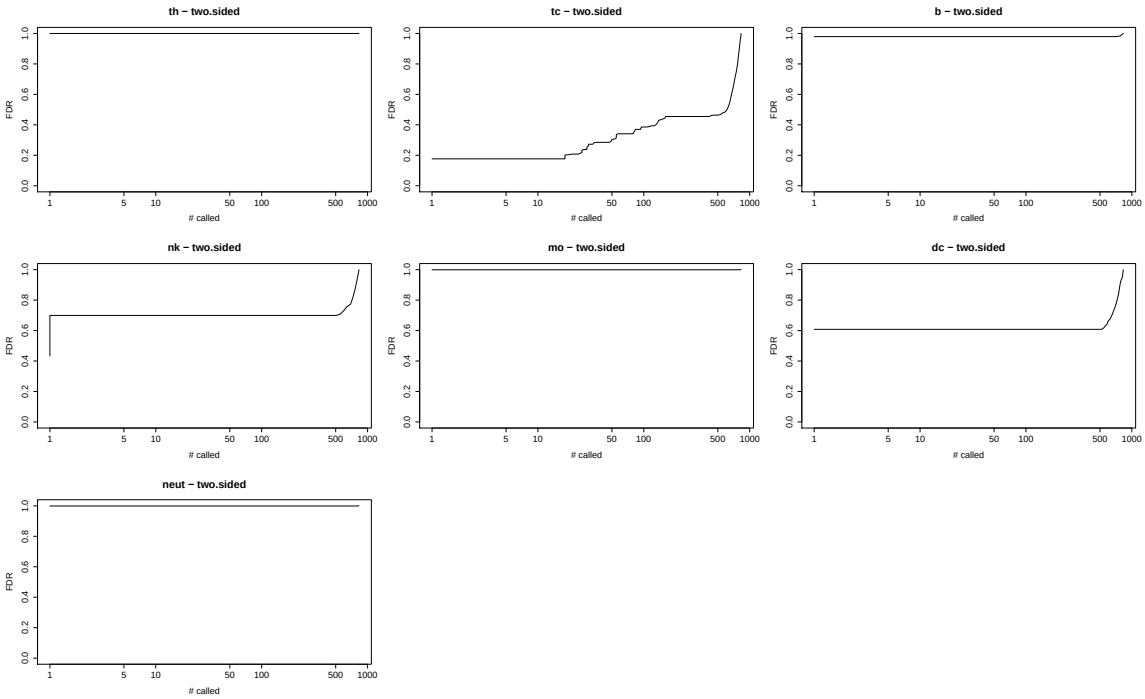


(a) **Heatmap of differentially expressed probes, HIV-1 uninfected**



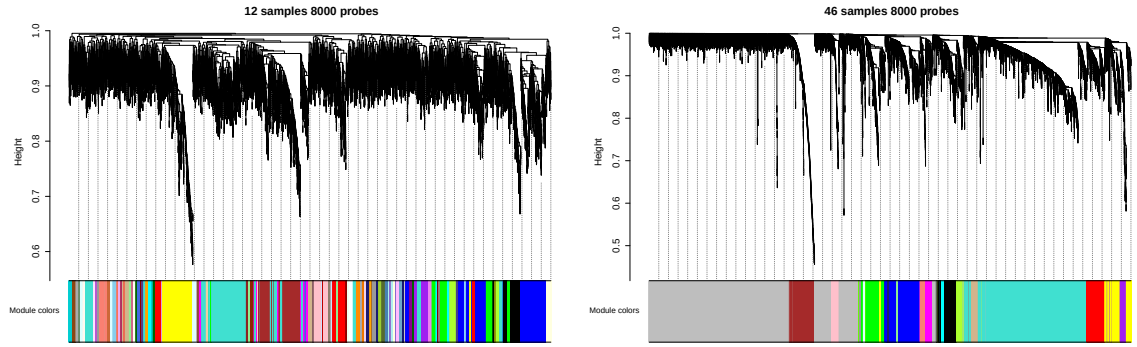
(b) **Heatmap of differentially expressed probes, HIV-1 infected**

Figure 10.5: **Results for contrast *TB site* in contexts *HIV negative* and *HIV positive*.** The heatmaps show good separation of samples based on site of TB infection. Few or no probes for significantly differentially expressed after Benjamini Hochberg correction, but despite this the top 300 differentially expressed probes (based on ranking by log-odds of differential expression) show good separation of the samples. The question of whether we are seeing a batch effect (the samples were collected in different studies) or a true biological effect remains currently an open one. Values are row-scaled, and range from low (blue) to intermediate (yellow) to high (red). Samples are clustered using Spearman rank correlation, and probes are clustered using Pearson correlation. Sample colours: orange: *PTB*, red: *TB-PC*. Plot titles reflect the number of included probes.



(a) **False discovery rate plot HIV-1 infected**

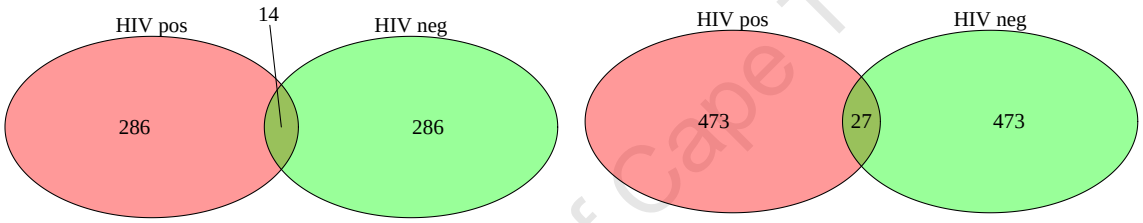
Figure 10.6: **Cell-type specific differential expression.** Only the result for HIV-1 infected individuals is shown, as the HIV-1 uninfected group was too limited in sample numbers for meaningful analysis.



(a) **Module clustering HIV neg**

(b) **Module clustering HIV pos**

Figure 10.7: **Module clustering.**



(a) **Overlap of top 300 differentially expressed probes** (b) **Overlap of top 500 differentially expressed probes**

Figure 10.8: **Overlap of differentially expressed probes.**

10.3 Question 3: Latent TB

This subsection addresses the question of whether LTBI leaves a detectable signal in the blood transcriptome relative to no prior TB sensitisation.

10.3.1 Included patients and samples

Table 10.5 summarises the variables age, sex and CD4 count across two groups of study subjects (HIV-1 uninfected and -infected) as well as within each of the two groups across two groups defined by the contrast variable *LTBI vs healthy*. The two groups, as well as the two comparator groups within each group were matched for age. Both comparator groups within each set were also matched for sex, but the two sets differed significantly w.r.t. this variable; no male patients were included in the HIV-1 infected group, whereas both men and women were included in the HIV negative group. Finally, assessment for matching for CD4 count was only possible in the HIV-1 infected group (where the healthy and LTBI groups were matched for CD4 count), as CD4 count data was unavailable for the HIV-1 uninfected individuals. In summary, some bias due to sex may be expected when comparing the results of the two analyses.

10.3.2 Overall results

Table 10.6 lists the results of the analysis for the two datasets of interest. Once again, few to no probes (0 and 1, respectively) were significantly differentially expressed after multiple testing correction. Deconvolution and cell-specific differential expression was successful for both datasets, as was module detection.

10.3.3 Results for context HIV: (datasets: Blood, HIV negative and Blood, HIV positive)

With selection of the top 300 differentially expressed probes, both HIV-1 uninfected and uninfected input data sets are separated perfectly into two clusters, suggesting that the probes are biologically meaningful.

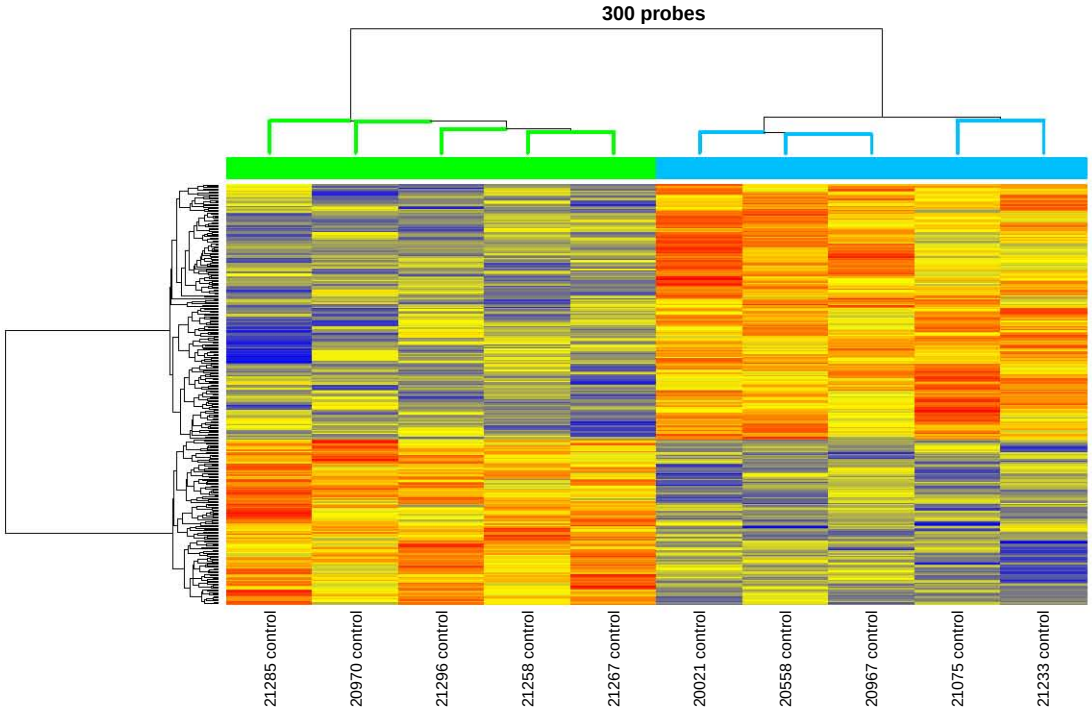
Cell-type specific differential expression in HIV-1 uninfected individuals only exhibited a very weak signal in natural killer cells and neutrophils, while in HIV-1 infected individuals a very strong

Table 10.5: **Clinical characteristics: Healthy vs LTBI.** Two-way comparison of demographic variables and CD4 count between and with groups of study participants. P-values less than 0.05 are regarded as significant.

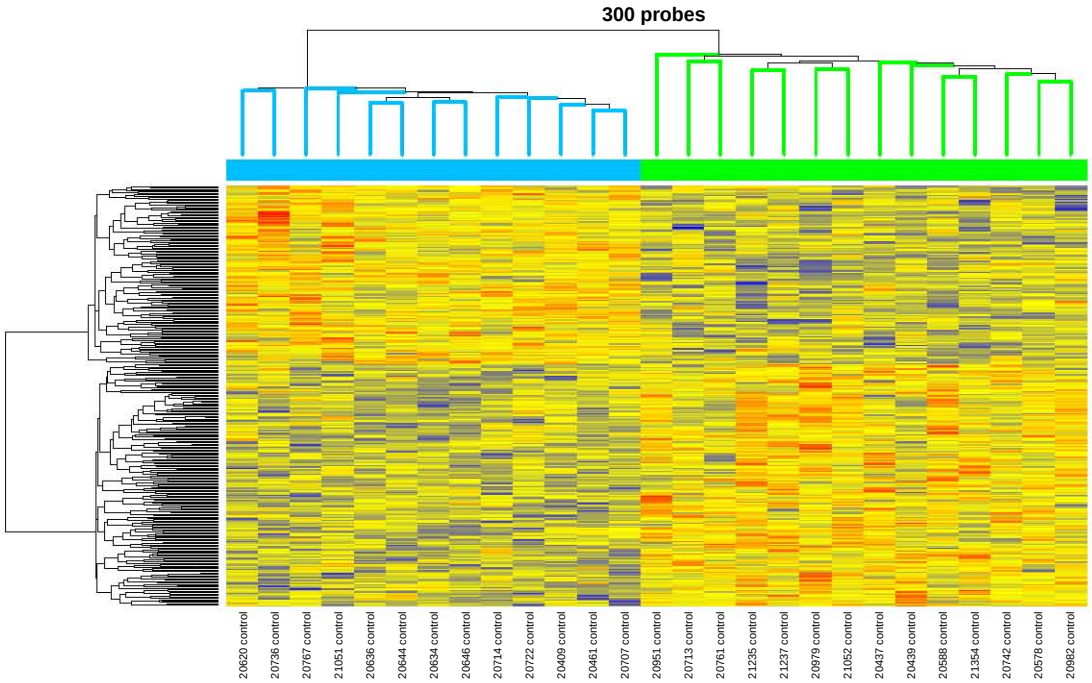
	Blood HIVpos Healthy-LTBI			Blood HIVneg Healthy-LTBI			Pval	Test
Age	Median	LQ	UQ	Median	LQ	UQ	Pval	Test
age_All	29.5	25.75	37.3	29.9	22.98	34.2	0.584	Kruskal-Wallis rank
age_ CON_LTBI	29.5	25.5	35.3	29.1	22.7	34.7	0.693	Kruskal-Wallis rank
age_ CON_healthy	29.5	26.08	39.47	30.7	23.8	32.7	0.711	Kruskal-Wallis rank
age_P value	0.544			0.917				
age_Test	Kruskal-Wallis rank			Kruskal-Wallis rank				
Sex	ratio F/M	Female	Male	ratio F/M	Female	Male	Pval	test
sex_All	Inf	27	0	0.250	2	8	0.000	2-sample test for eq
sex_ CON_LTBI	Inf	13	0	0.250	1	4	0.002	2-sample test for eq
sex_ CON_healthy	Inf	14	0	0.250	1	4	0.002	2-sample test for eq
sex_P value	NaN			1.000				
sex_Test	2-sample test for eq			2-sample test for eq				
CD4	Median	LQ	UQ	Median	LQ	UQ	Pval	Test
CD4_All	333	307.5	467				NA	NA
CD4_ CON_LTBI	333	316	416				NA	NA
CD4_ CON_healthy	372	273	542.5				NA	NA
CD4_P value	0.846			NA				
CD4_Test	Kruskal-Wallis rank			NA				

Table 10.6: **Overall results: Healthy vs LTBI.** Output of the analysis script, showing the numbers of included samples, and statistics for differential expression, deconvolution and gene co-expression network analysis. P-values less than 0.05 are regarded as significant.

Question 3: Latent TB infection		
Contrast: LTBI vs. Healthy		
Dataset	1	2
Context	Blood, HIV pos	Blood, HIV neg
N: LTBI	13	5
N: Healthy	14	5
Analysis 1: Differential expression		
N significant probes (unadjusted)	1329	2111
N significant probes (BH adjusted)	0	1
N probes used (heatmaps, csDE)	300	300
Analysis 2: Deconvolution		
Deconvolution successful	yes	yes
N of N significant for contrast (BH)	0 of 12	0 of 13
N of N PBMC significant for contrast (BH)	0 of 11	0 of 12
Cell-specific DE successful	yes	yes
Number of cell types	7	4
N cell types that reach FDR < 0.4	2	0
Analysis 3: WGCNA		
R^2 cutoff for scale-free approximation	0.8	0.8
Soft threshold power	3	10
Number of modules	21	32
Probes assigned to modules (N, %)	7182 (89.8)	6783 (84.8)
GO: Entrez IDs submitted	3900	4201
GO: Entrez IDs mapped	2573	2805
N (%) modules cor GS/MM sig	8 (38)	21 (65)
N (%) modules sig for contrast (BH)	20 (95)	24 (75)



(a) **Heatmap of differentially expressed probes, HIV-1 uninfected**



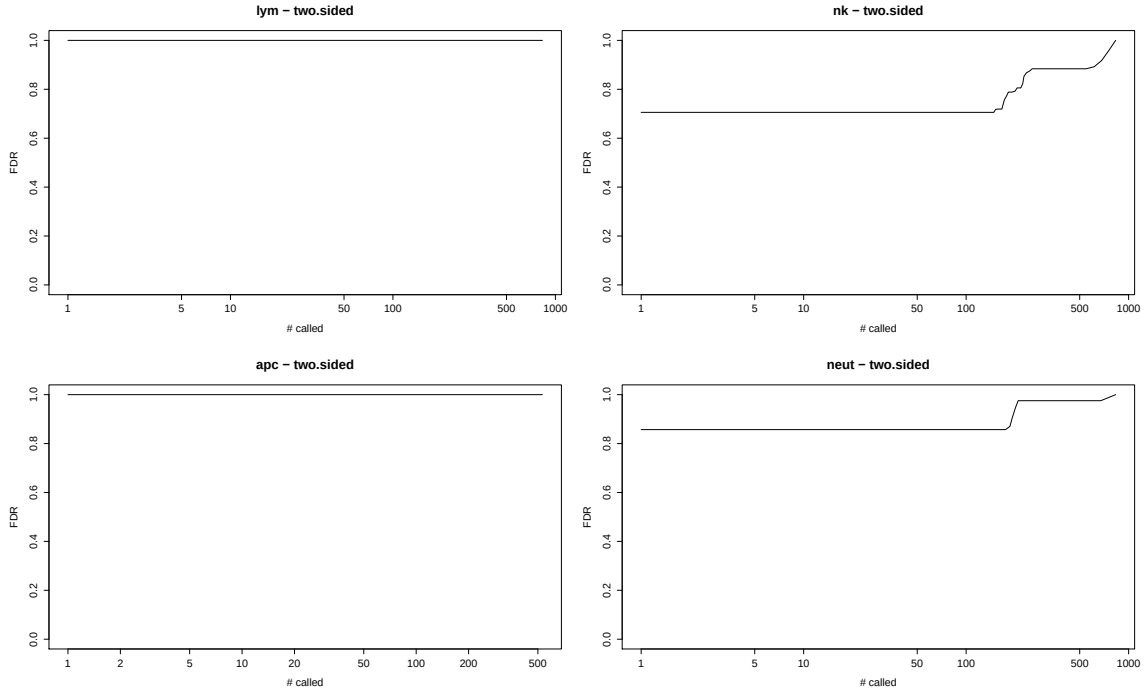
(b) **Heatmap of differentially expressed probes, HIV-1 infected**

Figure 10.9: **Results for contrast *LTBI* in contexts *HIV negative* and *HIV positive.*** Values are row-scaled, and range from low (blue) to intermediate (yellow) to high (red). Samples are clustered using Spearman rank correlation, and probes are clustered using Pearson correlation. Sample colours: green: *healthy*, blue: *LTBI*. Plot titles reflect the number of included probes.

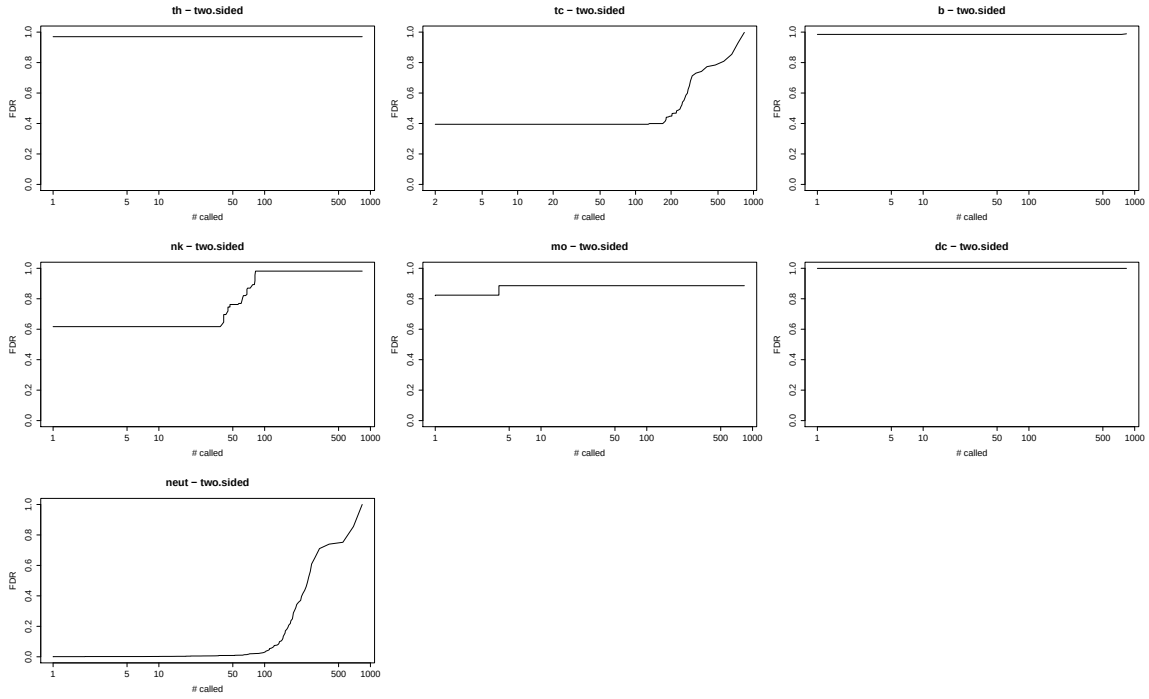
neutrophil signal combined with weaker signals for CD8 T cells and NK cells are evident.

Almost all probes were assigned to modules in both cases.

Once again, the top 300 differentially expressed probes hardly overlap (3 out of 300), again highlighting the possibility that HIV-1 infection in the context of LTBI drives the differential expression seen.

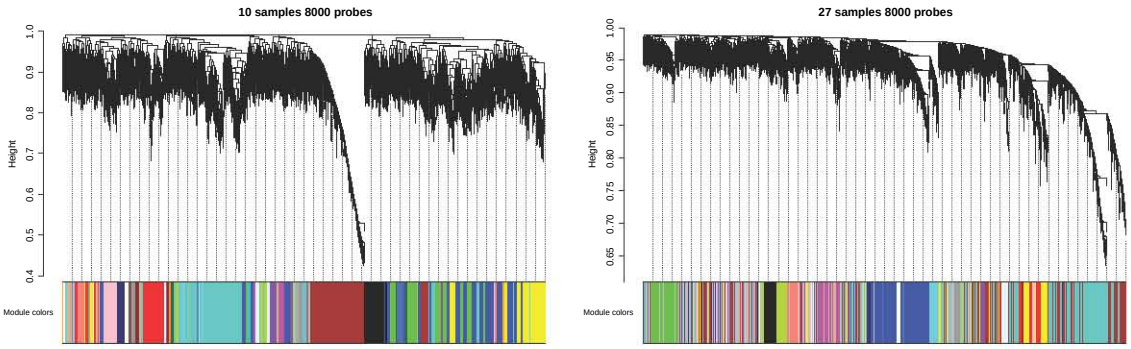


(a) **False discovery rate plot HIV-1 uninfected**



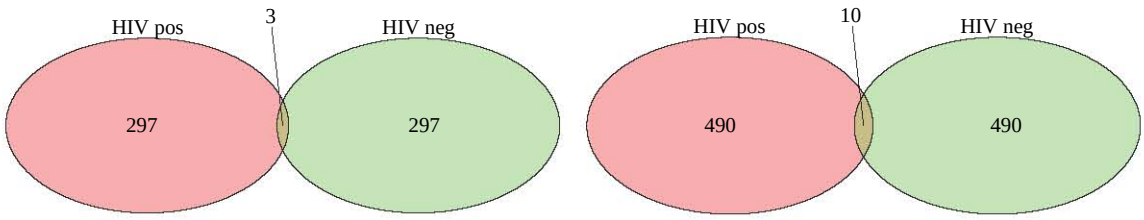
(b) **False discovery rate plot HIV-1 infected**

Figure 10.10: **Cell-type specific differential expression.**



(a) **Module clustering HIV neg** (b) **Module clustering HIV pos**

Figure 10.11: **Module clustering.**



(a) **Overlap of top 300 differentially expressed probes** (b) **Overlap of top 500 differentially expressed probes**

Figure 10.12: **Overlap of differentially expressed probes.**

10.4 Question 4: Haemodynamic phenotypes in TB-PC

The final question relating to tuberculosis deals with haemodynamic phenotypes. Here I examine the contrast "effusive TB-PC vs. effusive-constrictive TB-PC". Here I compare the results of this analysis for two compartments: blood and pericardial fluid, the site of disease. It is anticipated that the signal for effusive-constrictive disease, if any, is stronger in the pericardial fluid compartment.

10.4.1 Included patients and samples

As the comparison focuses on matched blood and pericardial fluid samples, all comparisons across the two datasets are not significant, as they consider the same study subjects. In addition, the samples are matched for age, sex and CD4 count across the contrast variable. See Table 10.7 for details.

Table 10.7: **Clinical characteristics: Effusive vs Effusive-constrictive pericarditis.** Two-way comparison of demographic variables and CD4 count between and with groups of study participants. P-values less than 0.05 are regarded as significant.

	Blood TBPC HIVPos Eff-EC			Fluid TBPC HIVPos Eff-EC			Pval	Test
Age	Median	LQ	UQ	Median	LQ	UQ	Pval	Test
age_All	32.14	27.91	36.72	32.14	27.91	36.72	1.000	Kruskal-Wallis rank
age_ EC	33.08	32.92	38.75	33.08	32.92	38.75	1.000	Kruskal-Wallis rank
age_ Eff	30.12	27.51	33.7	30.12	27.51	33.7	1.000	Kruskal-Wallis rank
age_P value	0.184			0.184				
age_Test	Kruskal-Wallis rank			Kruskal-Wallis rank				
Sex	ratio F/M	Female	Male	ratio F/M	Female	Male	Pval	test
sex_All	0.818	9	11	0.818	9	11	1.000	2-sample test for eq
sex_ EC	0.800	4	5	0.800	4	5	1.000	2-sample test for eq
sex_ Eff	0.833	5	6	0.833	5	6	1.000	2-sample test for eq
sex_P value	1.000			1.000				
sex_Test	2-sample test for eq			2-sample test for eq				
CD4	Median	LQ	UQ	Median	LQ	UQ	Pval	Test
CD4_All	98.5	62.5	192.2	98.5	62.5	192.2	1.000	Kruskal-Wallis rank
CD4_ EC	105	95	223	105	95	223	1.000	Kruskal-Wallis rank
CD4_ Eff	86	49	166.5	86	49	166.5	1.000	Kruskal-Wallis rank
CD4_P value	0.171			0.171				
CD4_Test	Kruskal-Wallis rank			Kruskal-Wallis rank				

10.4.2 Overall results

No probes were significantly differentially expressed after multiple testing correction, so again the top 300 differentially expressed probes ranked by log-odds for differential expression were used for downstream analyses and visualisations. Deconvolution, cell-type specific differential expression analysis and module detection were performed successfully. Table 10.8 lists the main results of this analysis.

Table 10.8: **Overall results: Effusive vs Effusive-constrictive pericarditis.** Output of the analysis script, showing the numbers of included samples, and statistics for differential expression, deconvolution and gene co-expression network analysis. P-values less than 0.05 are regarded as significant.

Question 4: Haemodynamic phenotype		
Contrast: Effusive-constrictive pericarditis (EC) vs. Effusive pericarditis (Eff)		
Dataset	1	2
Context	Blood, TB-PC, HIV pos	Fluid, TB-PC, HIV pos
N: EC	9	9
N: Eff	11	11
Analysis 1: Differential expression		
N significant probes (unadjusted)	1195	1269
N significant probes (BH adjusted)	0	0
N probes used (heatmaps, csDE)	300	300
Analysis 2: Deconvolution		
Deconvolution successful	yes	yes
N of N significant for contrast (BH)	0 of 13	0 of 13
N of N PBMC significant for contrast (BH)	0 of 12	0 of 12
Cell-specific DE successful	yes	yes
Number of cell types	7	7
N cell types that reach FDR < 0.4	4	0
Analysis 3: WGCNA		
R^2 cutoff for scale-free approximation	3	14
Soft threshold power	0.8	0.8
Number of modules	34	12
Probes assigned to modules (N, %)	7808 (97.6)	2250 (28.1)
GO: Entrez IDs submitted	3759	3912
GO: Entrez IDs mapped	2452	2574
N (%) modules cor GS/MM sig	18 (53)	3 (25)
N (%) modules sig for contrast (BH)	33 (97)	1 (8)

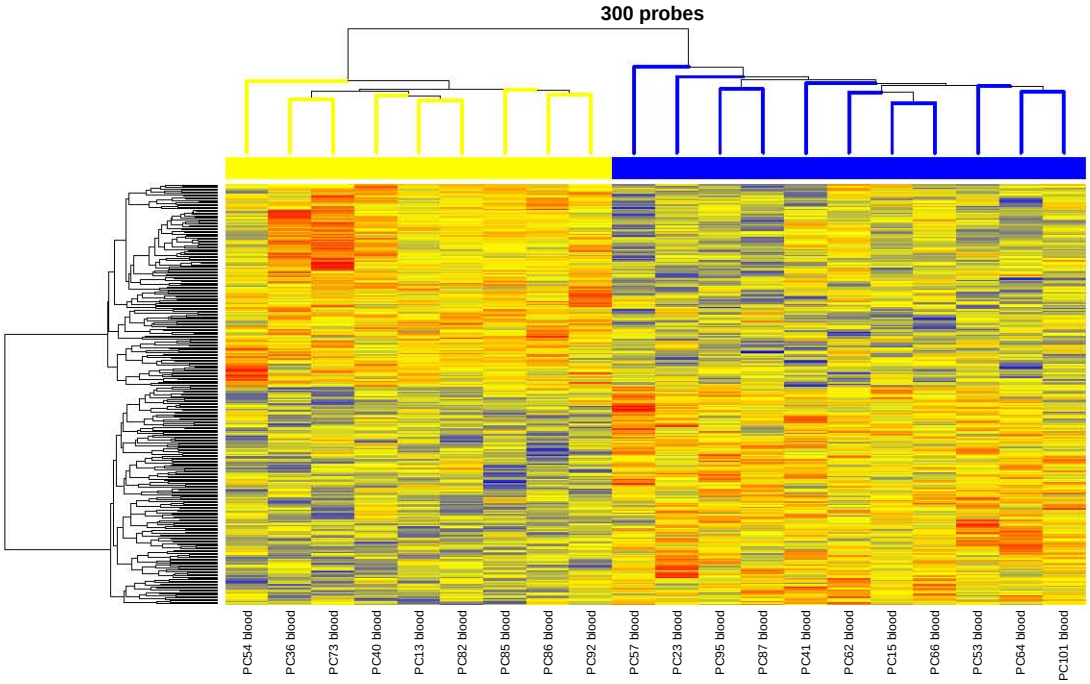
10.4.3 Results for context: Compartment (datasets: TB-PC HIVpos Blood, TB-PC HIVpos Fluid)

The heatmaps shown in Figure 10.13 show that the top 300 differentially expressed probes cluster their respective input data perfectly.

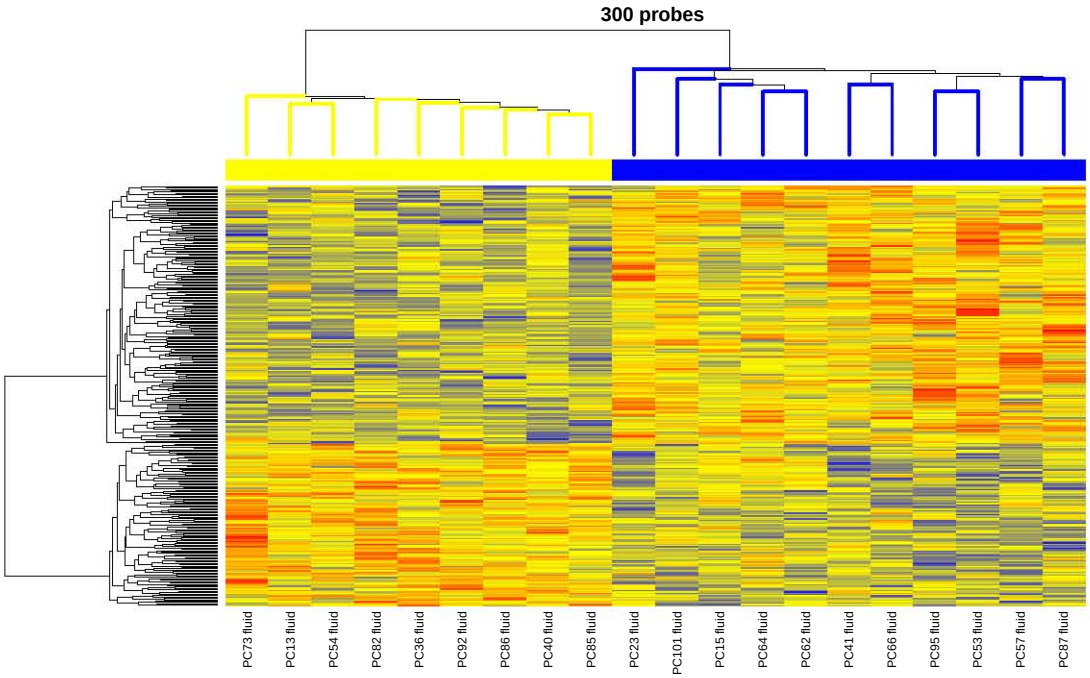
Cell-type specific differential expression yielded stronger signals in blood (for CD4 T cells, B cells NK cells, monocytes and dendritic cells) than in pericardial fluid (for neutrophils and CD8 T cells). This interesting finding warrants further investigation.

Module clustering is shown in Figure 10.15. Of interest is the fact that most probes were not assigned to any module.

As seen previously, the overlap of differentially expressed probes is very small when comparing blood and pericardial fluid. Blood appears to be a very poor mirror for events at the site of disease.

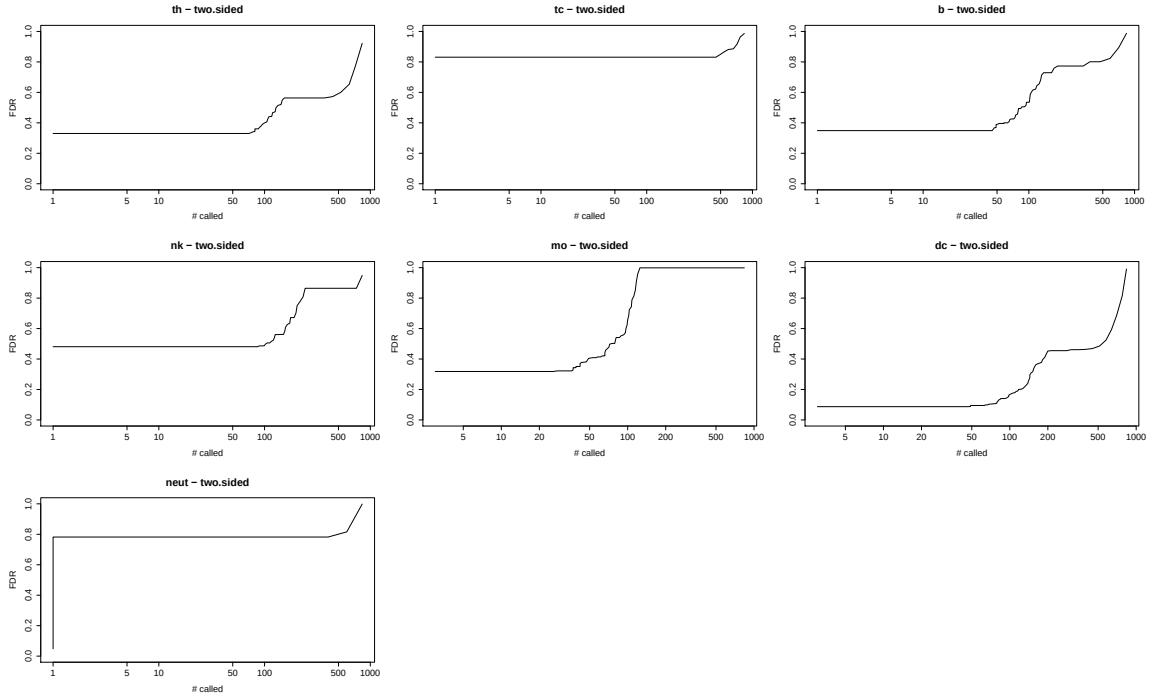


(a) **Heatmap of differentially expressed probes, blood**

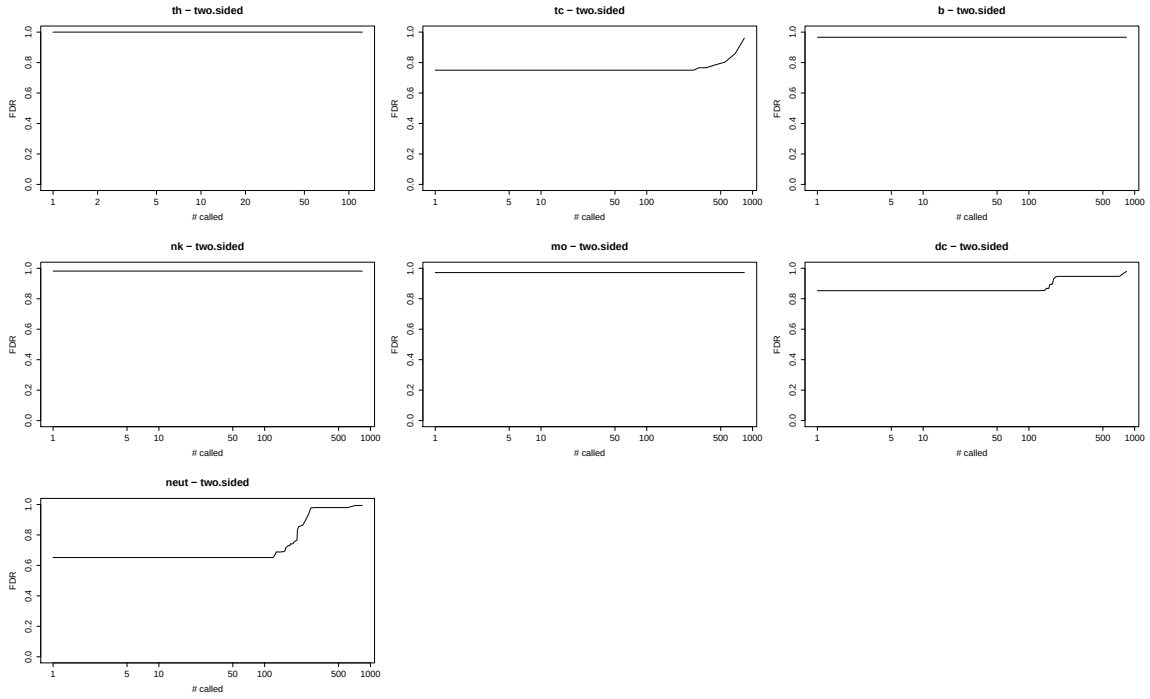


(b) **Heatmap of differentially expressed probes, pericardial fluid**

Figure 10.13: **Results for contrast *Haemodynamic phenotype* in contexts *Blood* and *Fluid.*** Values are row-scaled, and range from low (blue) to intermediate (yellow) to high (red). Samples are clustered using Spearman rank correlation, and probes are clustered using Pearson correlation. Sample colours: yellow: *effusive-constrictive phenotype*, blue: *effusive phenotype.* Plot titles reflect the number of included probes.

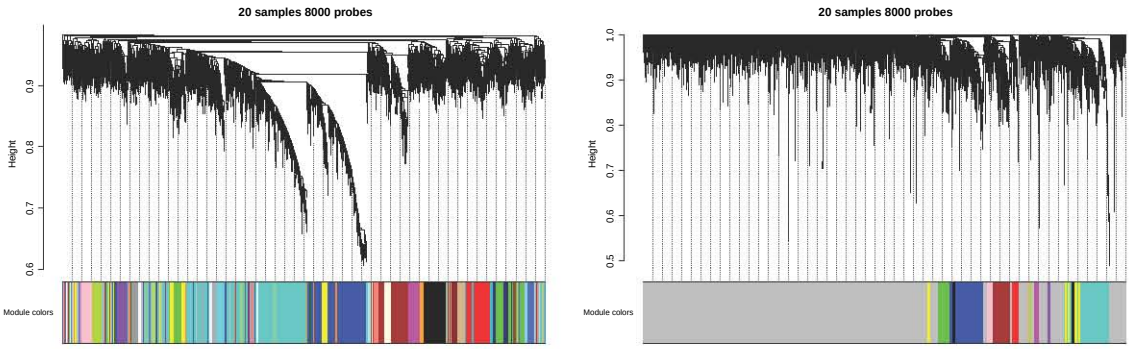


(a) **False discovery rate plot blood**



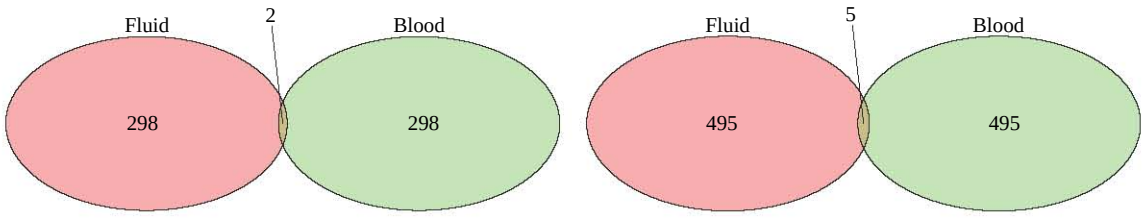
(b) **False discovery rate plot pericardial fluid**

Figure 10.14: **Cell-type specific differential expression.**



(a) **Module clustering Blood** (b) **Module clustering Fluid**

Figure 10.15: **Module clustering.**



(a) **Overlap of top 300 differentially expressed probes** (b) **Overlap of top 500 differentially expressed probes**

Figure 10.16: **Overlap of differentially expressed probes.**

10.5 Pathway analysis

Pathway analysis was performed as described in Section 3.8.8. Tables 10.9, 10.10 and 10.11 list the pathways found to be significantly differentially regulated in active tuberculosis in HIV-1 uninfected and -infected groups, respectively. Figures 10.18 and 10.19show heatmaps for the differentially regulated pathways in tuberculosis cases.

Table 10.9: **Top pathways in TB (HIV-1 uninfected).** Positive results refer to upregulation of the pathway in active tuberculosis, and negative numbers refer to downregulation. Colours: *green* = also downregulated in HIV-TB, *red* = also upregulated in HIV-TB, *yellow* = downregulated in HIV-TB. The *Mean logFC* value indicates the extent to which the pathway is up- or downregulated by providing the $\log_2$-fold change value.

Pathway	Mean logFC
hsa00190 Oxidative phosphorylation	2.356
hsa03050 Proteasome	1.573
hsa04145 Phagosome	1.365
hsa04610 Complement and coagulation cascades	1.256
hsa04260 Cardiac muscle contraction	1.073
hsa04142 Lysosome	0.930
hsa04620 Toll-like receptor signaling pathway	0.894
hsa04130 SNARE interactions in vesicular transport	0.888
hsa04612 Antigen processing and presentation	0.764
hsa03060 Protein export	0.773
hsa04660 T cell receptor signaling pathway	-1.418
hsa04630 Jak-STAT signaling pathway	-1.148
hsa04310 Wnt signaling pathway	-1.133
hsa04070 Phosphatidylinositol signaling system	-1.106
hsa04662 B cell receptor signaling pathway	-0.937
hsa04144 Endocytosis	-0.884
hsa04012 ErbB signaling pathway	-0.853
hsa04720 Long-term potentiation	-0.827
hsa00562 Inositol phosphate metabolism	-0.818
hsa04742 Taste transduction	-0.810
hsa00532 Glycosaminoglycan biosynthesis - chondroitin sulfate	-0.779
hsa04010 MAPK signaling pathway	-0.765
hsa04330 Notch signaling pathway	-0.766
hsa04960 Aldosterone-regulated sodium reabsorption	-0.748
hsa00514 Other types of O-glycan biosynthesis	-0.740
hsa00563 Glycosylphosphatidylinositol(GPI)-anchor biosynthesis	-0.726
hsa03040 Spliceosome	-0.720
hsa03013 RNA transport	-0.712
hsa04910 Insulin signaling pathway	-0.706
hsa04520 Adherens junction	-0.689
hsa04110 Cell cycle	-0.678
hsa00270 Cysteine and methionine metabolism	-0.662
hsa04062 Chemokine signaling pathway	-0.640

Table 10.10: **Upregulated pathways in TB (HIV-1 infected).** Colours: *red* = also upregulated in TB without HIV

Pathway	Mean logFC
hsa04130 SNARE interactions in vesicular transport	1.219
hsa04620 Toll-like receptor signaling pathway	1.136
hsa04380 Osteoclast differentiation	1.133
hsa00190 Oxidative phosphorylation	1.096
hsa04610 Complement and coagulation cascades	1.082
hsa00564 Glycerophospholipid metabolism	0.968
hsa04140 Regulation of autophagy	0.923
hsa00760 Nicotinate and nicotinamide metabolism	0.898
hsa04966 Collecting duct acid secretion	0.887
hsa00600 Sphingolipid metabolism	0.711
hsa00770 Pantothenate and CoA biosynthesis	0.715
hsa04740 Olfactory transduction	0.663
hsa04976 Bile secretion	0.623
hsa02010 ABC transporters	0.622
hsa00601 Glycosphingolipid biosynthesis - lacto and neolacto series	0.623
hsa04142 Lysosome	0.622
hsa00512 Mucin type O-Glycan biosynthesis	0.604
hsa04920 Adipocytokine signaling pathway	0.570
hsa04145 Phagosome	0.561
hsa03320 PPAR signaling pathway	0.532
hsa00980 Metabolism of xenobiotics by cytochrome P450	0.511
hsa04260 Cardiac muscle contraction	0.509
hsa04614 Renin-angiotensin system	0.484
hsa00982 Drug metabolism - cytochrome P450	0.462
hsa00900 Terpenoid backbone biosynthesis	0.456
hsa03022 Basal transcription factors	0.443
hsa00480 Glutathione metabolism	0.422
hsa04975 Fat digestion and absorption	0.416
hsa04210 Apoptosis	0.409
hsa00531 Glycosaminoglycan degradation	0.370
hsa00590 Arachidonic acid metabolism	0.352
hsa00910 Nitrogen metabolism	0.344
hsa04710 Circadian rhythm - mammal	0.333
hsa04621 NOD-like receptor signaling pathway	0.314
hsa00380 Tryptophan metabolism	0.313

Table 10.11: **Downregulated pathways in TB (HIV-1 infected).** Colours: green = also down in TB without HIV, yellow = upregulated in TB without HIV

Pathway	Mean logFC
hsa03040 Spliceosome	-1.779
hsa03013 RNA transport	-1.501
hsa04660 T cell receptor signaling pathway	-1.384
hsa00970 Aminoacyl-tRNA biosynthesis	-1.259
hsa00290 Valine, leucine and isoleucine biosynthesis	-1.220
hsa03008 Ribosome biogenesis in eukaryotes	-1.154
hsa00270 Cysteine and methionine metabolism	-1.084
hsa03030 DNA replication	-1.079
hsa04110 Cell cycle	-1.029
hsa03018 RNA degradation	-1.024
hsa04612 Antigen processing and presentation	-0.851
hsa00620 Pyruvate metabolism	-0.830
hsa04520 Adherens junction	-0.822
hsa00020 Citrate cycle (TCA cycle)	-0.793
hsa04662 B cell receptor signaling pathway	-0.753
hsa00640 Propanoate metabolism	-0.747
hsa00532 Glycosaminoglycan biosynthesis - chondroitin sulfate	-0.729
hsa00010 Glycolysis / Gluconeogenesis	-0.707
hsa04514 Cell adhesion molecules (CAMs)	-0.700
hsa00280 Valine, leucine and isoleucine degradation	-0.687
hsa04672 Intestinal immune network for IgA production	-0.668
hsa04310 Wnt signaling pathway	-0.651
hsa00250 Alanine, aspartate and glutamate metabolism	-0.644
hsa03010 Ribosome	-0.665
hsa00670 One carbon pool by folate	-0.509
hsa00310 Lysine degradation	-0.478
hsa04114 Oocyte meiosis	-0.476
hsa03015 mRNA surveillance pathway	-0.472
hsa03430 Mismatch repair	-0.468
hsa04720 Long-term potentiation	-0.465
hsa00051 Fructose and mannose metabolism	-0.457
hsa04062 Chemokine signaling pathway	-0.447
hsa04070 Phosphatidylinositol signaling system	-0.435
hsa04540 Gap junction	-0.413
hsa00030 Pentose phosphate pathway	-0.418
hsa00410 beta-Alanine metabolism	-0.406
hsa00350 Tyrosine metabolism	-0.392
hsa04530 Tight junction	-0.385
hsa04330 Notch signaling pathway	-0.372
hsa04510 Focal adhesion	-0.365
hsa00360 Phenylalanine metabolism	-0.365
hsa04960 Aldosterone-regulated sodium reabsorption	-0.359
hsa00040 Pentose and glucuronate interconversions	-0.357
hsa04742 Taste transduction	-0.350
hsa00920 Sulfur metabolism	-0.354
hsa04640 Hematopoietic cell lineage	-0.346
hsa04744 Phototransduction	-0.335
hsa00630 Glyoxylate and dicarboxylate metabolism	-0.336
hsa04650 Natural killer cell mediated cytotoxicity	-0.322
hsa03410 Base excision repair	-0.312
hsa04350 TGF-beta signaling pathway	-0.306
hsa00340 Histidine metabolism	-0.303
hsa03450 Non-homologous end-joining	-0.300
hsa04722 Neurotrophin signaling pathway	-0.275

Figure 10.17 demonstrates the overlap in up- and downregulated pathways in active tuberculosis in HIV-1 infected and uninfected individuals. 7 and 15 pathways are up and downregulated in active TB, respectively, *regardless of HIV status*. Note that one pathway (hsa04612 Antigen processing and presentation) that is upregulated in HIV-1 uninfected individuals is downregulated in HIV-1 co-infection. It is clear that in addition to the pathways up- or downregulated in active tuberculosis in HIV-1 uninfected individuals, HIV-1 co-infection results in additional differential pathway regulation in the presence of active tuberculosis. It is unclear whether this association is causal, and if so, in which direction.

Broadly speaking, upregulated pathways are concerned with meeting increased energy requirements in tuberculosis (oxidative phosphorylation), phagocytosis and downstream events like vesicular transport (phagosome, lysosome) and innate immune responses (Toll-like receptor signalling and complement pathways). In contrast, downregulated pathways may indicate possible mechanisms by which *Mycobacterium tuberculosis* evades internalisation and killing (e.g endocytosis, chemokine signalling) and subverts the host metabolic response to one more favourable to its survival (four metabolic pathways are downregulated). Most pathways up- or downregulated in HIV-1 uninfected individuals show the same pattern of regulation in HIV-1 infection (see Tables 10.10 and 10.11). Multiple additional pathways are also up- or downregulated in HIV-1 co-infected individuals, including additional metabolic pathways related to lipid metabolism, apoptosis and natural killer cell mediated cytotoxicity. The latter is of interest as HIV-1 has been shown to affect NK cell activation by Mycobacteria [161].

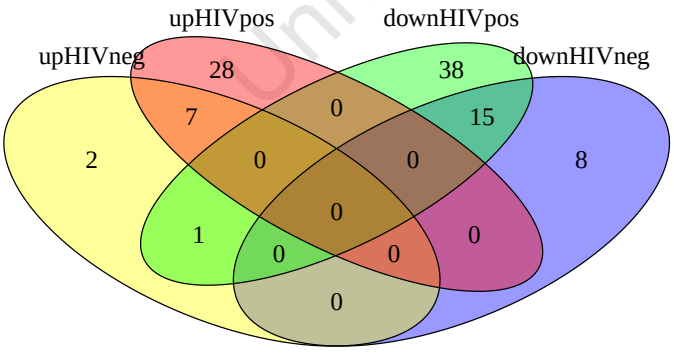


Figure 10.17: **Overlap of differentially regulated pathways in active tuberculosis stratified by HIV status.**

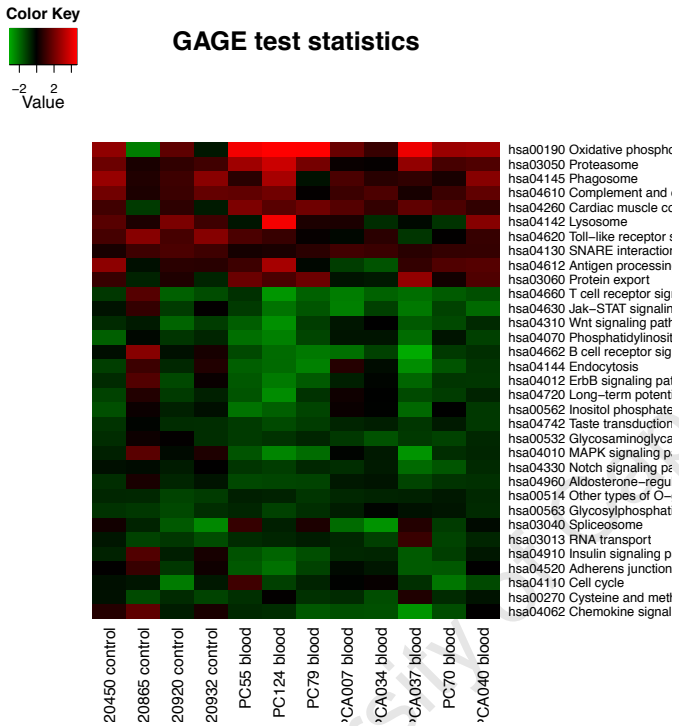


Figure 10.18: **Significant pathways in tuberculosis (HIV-1 uninfected).** The heatmap shows mean estimates for pathway expression for all active TB samples (not active TB samples are not shown). Colours: *red* = pathway upregulated; *green* = pathway downregulated. *(Note: Pathway names are cut off by the heatmap rendering algorithm; some manual editing of the source is required to fix this. The relevant tables provide the full pathway names)*

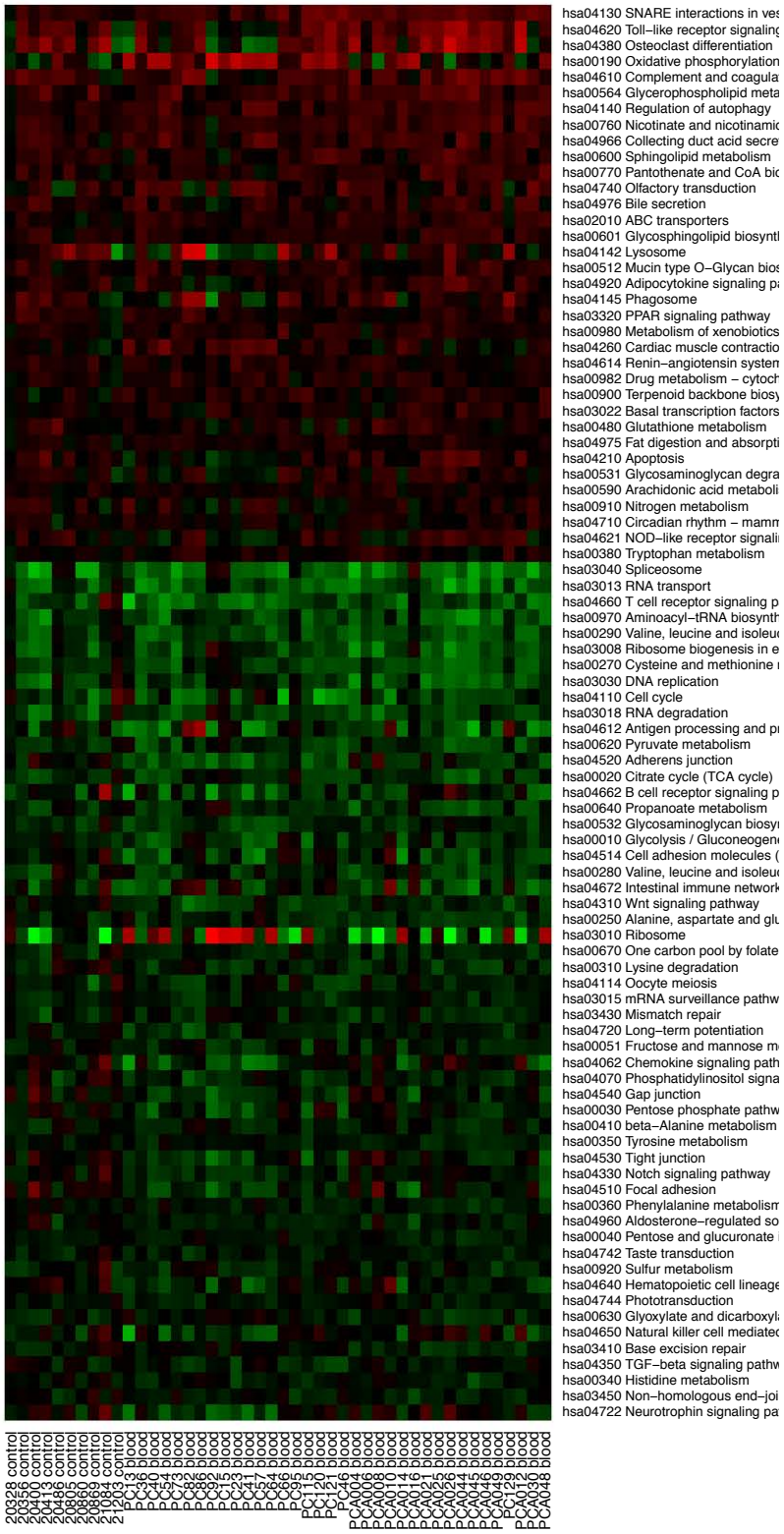
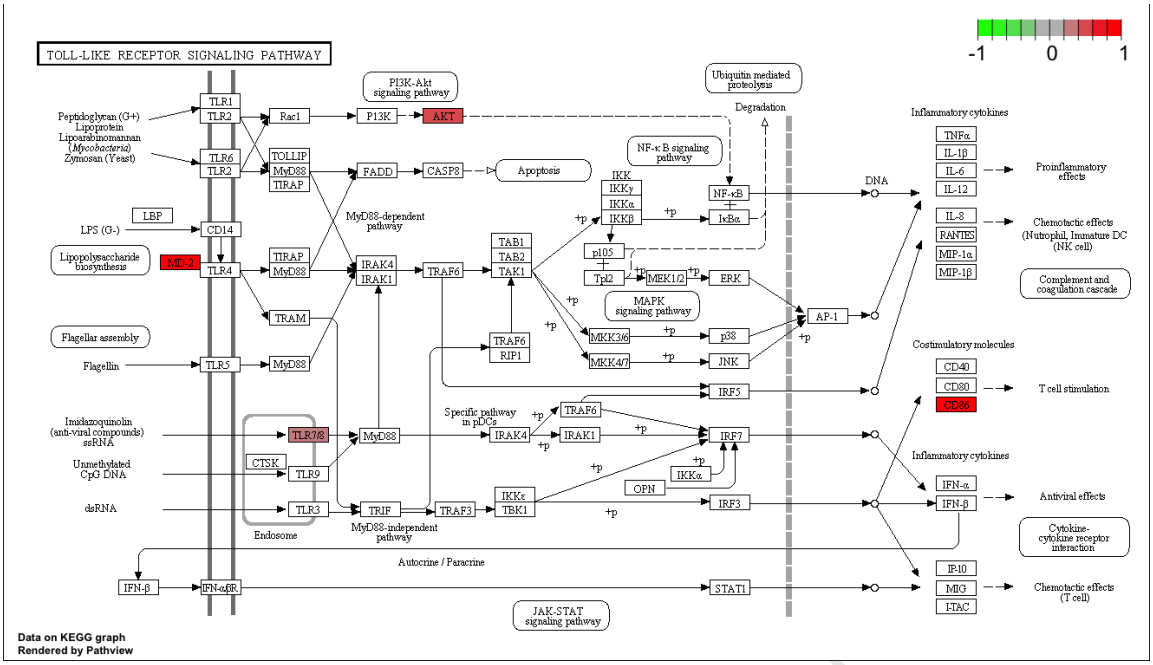


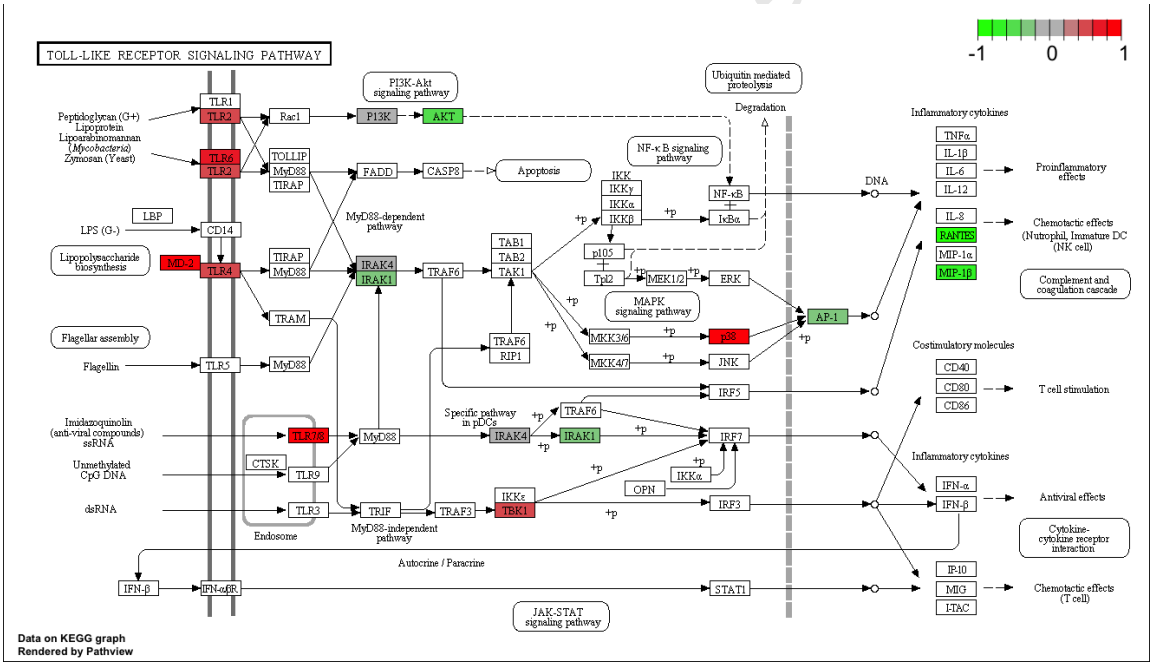
Figure 10.19: **Significant pathways in tuberculosis (HIV-1 infected).** The heatmap shows mean estimates for pathway expression for all active TB samples (not active TB samples are not shown). Colours: *red* = pathway upregulated; *green* = pathway downregulated.

10.5.1 Pathway visualisation

Figure 10.20 shows details of the Toll-like receptor pathway as an example of a pathway reported as upregulated in active tuberculosis in both HIV-1 uninfected and -infected individuals. While overall the pathway is reported as upregulated relative to controls, it appears that the biological effects of this upregulation may differ, as different individual genes in this pathway are over- or underexpressed. In Figure 10.21 we can see the effect of tuberculosis on the antigen processing and presentation pathway. In HIV-1 uninfected individuals, the MHC class I pathway is upregulated, while in HIV-1 infected individuals, the MHC class II pathway appears to be downregulated. This sheds some light on potential pathogenetic mechanisms in HIV-TB co-infection. Finally, Figure 10.22 shows the NK cell cytotoxicity pathway in blood for HIV-1 uninfected individuals. The set of top 100 probes called as differentially expressed in NK cells using cell-type specific differential expression analysis was intersected with the top 2000 differentially expressed probes in the standard differential expression analysis in order to provide logfold change estimates for the NK cell "specific" probes. Three of these probes are mapped to genes in the NK cell cytotoxicity pathway; two of these lie downstream of NKp30, NKp46 and FcγRIII receptors, suggesting that these receptors are involved in activating NK cells in tuberculosis. TNF-related apoptosis inducing ligand (TRAIL) is also upregulated in NK cells in tuberculosis, suggesting the effector pathway by which these NK cells may mediate their effect by inducing apoptosis in cells infected with *M. tuberculosis*. These results suggest numerous downstream experiments for future work.

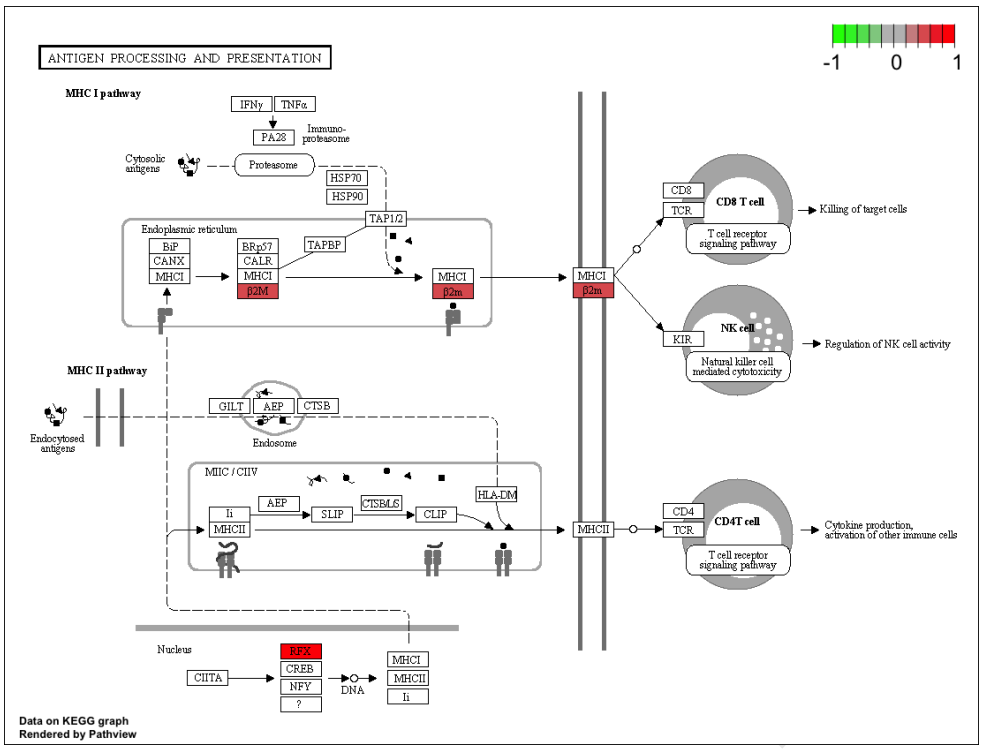


(a) **HIV-1 uninfected**

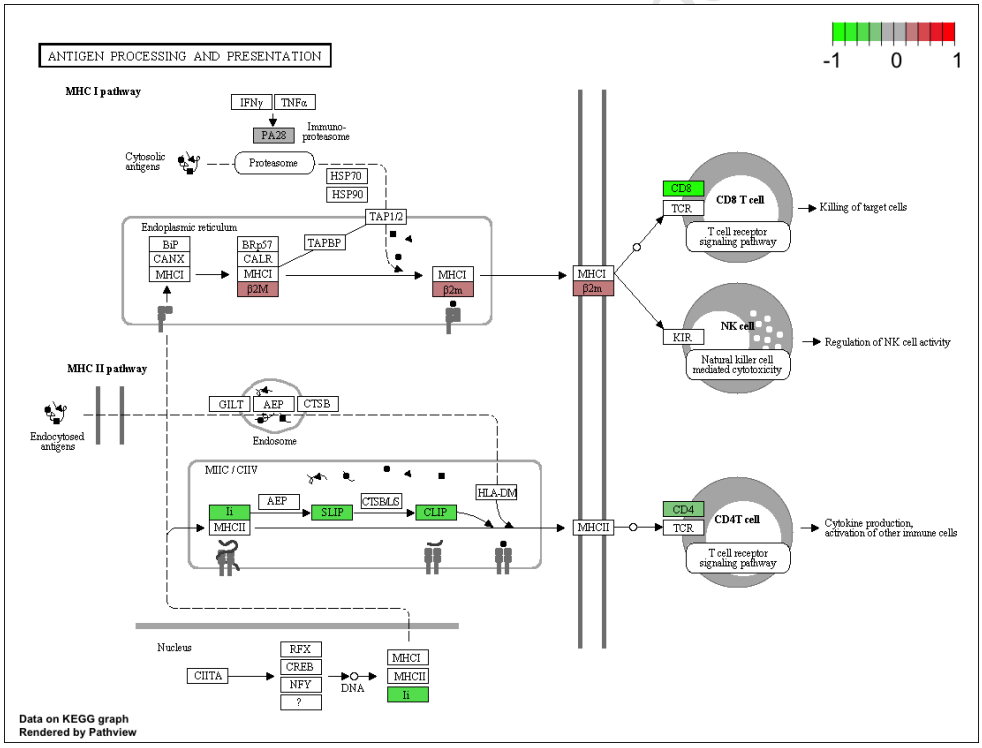


(b) **HIV-1 infected**

Figure 10.20: **Toll-like receptor pathway.** Canonical pathway *hsa04620* from KEGG, rendered by *pathview* software. The log-fold change estimates output by limma are overlaid, normalised to a range -1 to +1. Colours: *red* = upregulated in active tuberculosis; *green* = downregulated in active tuberculosis.



(a) **HIV-1 uninfected**



(b) **HIV-1 infected**

Figure 10.21: **Antigen processing and presentation pathway.** Canonical pathway *hsa04612* from KEGG, rendered by *pathview* software. The logfold change estimates output by limma are overlaid, normalised to a range -1 to +1. Colours: *red* = upregulated in active tuberculosis; *green* = downregulated in active tuberculosis.

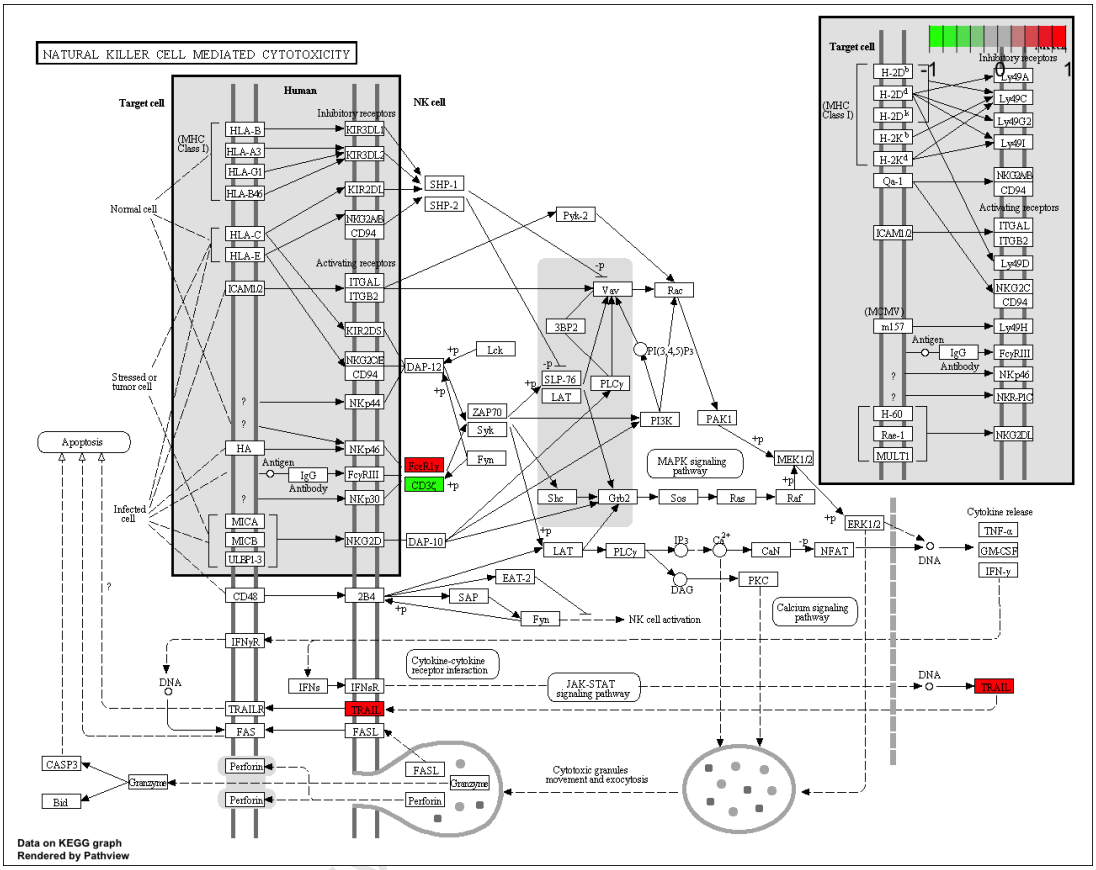


Figure 10.22: **NK cell cytotoxicity pathway (HIV-1 uninfected).** Canonical pathway *hsa04650*, rendered using *pathview*. Genes associated with differential expression in NK cells and shown to be differentially expressed globally in blood are highlighted in colour. Colours: *red* = upregulated in active tuberculosis; *green* = downregulated in tuberculosis. See text for interpretation.

11 Contrasts involving HIV infection status

Chapter overview

This Chapter deals with questions relating to the effect of HIV-1 infection in two contexts: not active tuberculosis and active tuberculosis.

11.1 Question 5: HIV (not active TB)

The effect of HIV-1 co-infection on tuberculosis is complex, and causal relationships are not always clear. In order to isolate the effect of HIV-1 infection from the effect of *M. tuberculosis* disease, I present an overview of the analysis of two datasets: *blood, LTBI* and *blood, healthy*.

11.1.1 Included patients and samples

Table 11.1 shows that the *healthy* and *LTBI* subsets of the class *not active TB* were matched for age, sex and CD4 count. Both datasets were individually matched for age for the contrast *HIV negative vs HIV positive*. In both cases, a higher proportion of female individuals constituted the HIV-1 infected class.

11.1.2 Overall results

The overall results (Table 11.2) show similar numbers of differentially expressed probes prior to multiple testing correction, with no or few (6) probes significant after multiple testing. This is likely due to the small number of included samples, and the results should be treated with reserve.

Table 11.1: **Clinical characteristics: HIV positive vs HIV negative (not active TB).** Two-way comparison of demographic variables and CD4 count between and with groups of study participants. P-values less than 0.05 are regarded as significant. This table includes data for the combined group ("not active TB"), LTBI and healthy.

	Blood not activeTB HIVpos-HIVneg			Blood LTBI HIVpos-HIVneg			Blood Healthy HIVpos-HIVneg			Pval	Test
Age	Median	LQ	UQ	Median	LQ	UQ	Median	LQ	UQ	Pval	Test
age_All	29.5	25	36.7	29.3	24.22	35.3	29.8	25.5	38.55	0.844	Kruskal-Wallis rank
age_ negative	29.9	22.98	34.2	29.1	22.7	34.7	30.7	23.8	32.7	0.994	Kruskal-Wallis rank
age_ positive	29.5	25.75	37.3	29.5	25.5	35.3	29.5	26.08	39.47	0.829	Kruskal-Wallis rank
age_P value	0.584			0.693			0.711				
age_Test	Kruskal-Wallis rank			Kruskal-Wallis rank			Kruskal-Wallis rank				
Sex	ratio F/M	Female	Male	ratio F/M	Female	Male	ratio F/M	Female	Male	Pval	test
sex_All	3.625	29	8	3.500	14	4	3.750	15	4	1.000	Fisher's Exact Test
sex_ negative	0.250	2	8	0.250	1	4	0.250	1	4	1.000	Fisher's Exact Test
sex_ positive	Inf	27	0	Inf	13	0	Inf	14	0	1.000	Fisher's Exact Test
sex_P value	0.000			0.002			0.002				
sex_Test	2-sample test for eq			2-sample test for eq			2-sample test for eq				
CD4	Median	LQ	UQ	Median	LQ	UQ	Median	LQ	UQ	Pval	Test
CD4_All	333	307.5	467	333	316	416	372	273	542.5	0.981	Kruskal-Wallis rank
CD4_ negative										NA	NA
CD4_ positive	333	307.5	467	333	316	416	372	273	542.5	0.981	Kruskal-Wallis rank
CD4_P value	NA			NA			NA				
CD4_Test	NA			NA			NA				

The top 300 differentially expressed probes based on log-likelihood of differential expression were used for subsequent analysis.

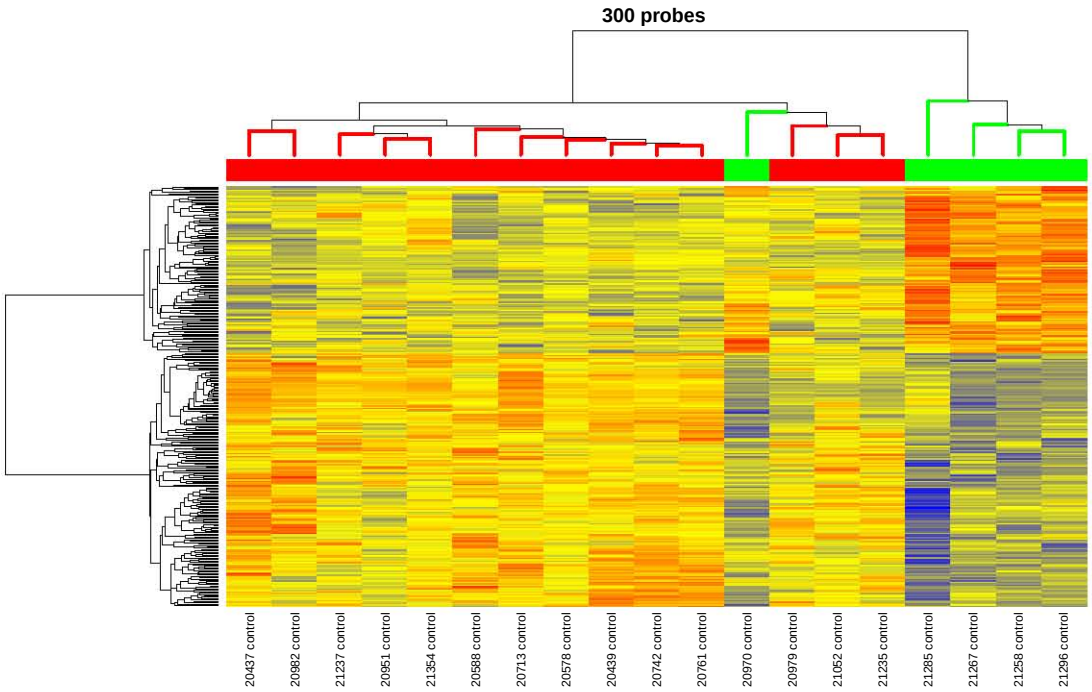
Table 11.2: **Overall results: HIV positive vs HIV negative (not active TB).** Output of the analysis script, showing the numbers of included samples, and statistics for differential expression, deconvolution and gene co-expression network analysis. P-values less than 0.05 are regarded as significant.

Question 5: HIV (no TB)		
Contrast: HIV neg vs. HIV pos		
Dataset	1	2
Context	Blood, LTBI	Blood, Healthy
N: HIV neg	5	5
N: HIV pos	13	14
Analysis 1: Differential expression		
N significant probes (unadjusted)	2299	3169
N significant probes (BH adjusted)	0	6
N probes used (heatmaps, csDE)	300	300
Analysis 2: Deconvolution		
Deconvolution successful	yes	yes
N of N significant for contrast (BH)	2 of 13	0 of 12
N of N PBMC significant for contrast (BH)	2 of 12	0 of 11
Cell-specific DE successful	yes	yes
Number of cell types	4	7
N cell types that reach FDR < 0.4	2	1
Analysis 3: WGCNA		
R^2 cutoff for scale-free approximation	0.8	0.8
Soft threshold power	6	7
Number of modules	23	15
Probes assigned to modules (N, %)	7968 (99.6)	6380 (79.75)
GO: Entrez IDs submitted	4114	4251
GO: Entrez IDs mapped	2720	2819
N (%) modules cor GS/MM sig	14 (61)	15 (100)
N (%) modules sig for contrast (BH)	23 (100)	15 (100)

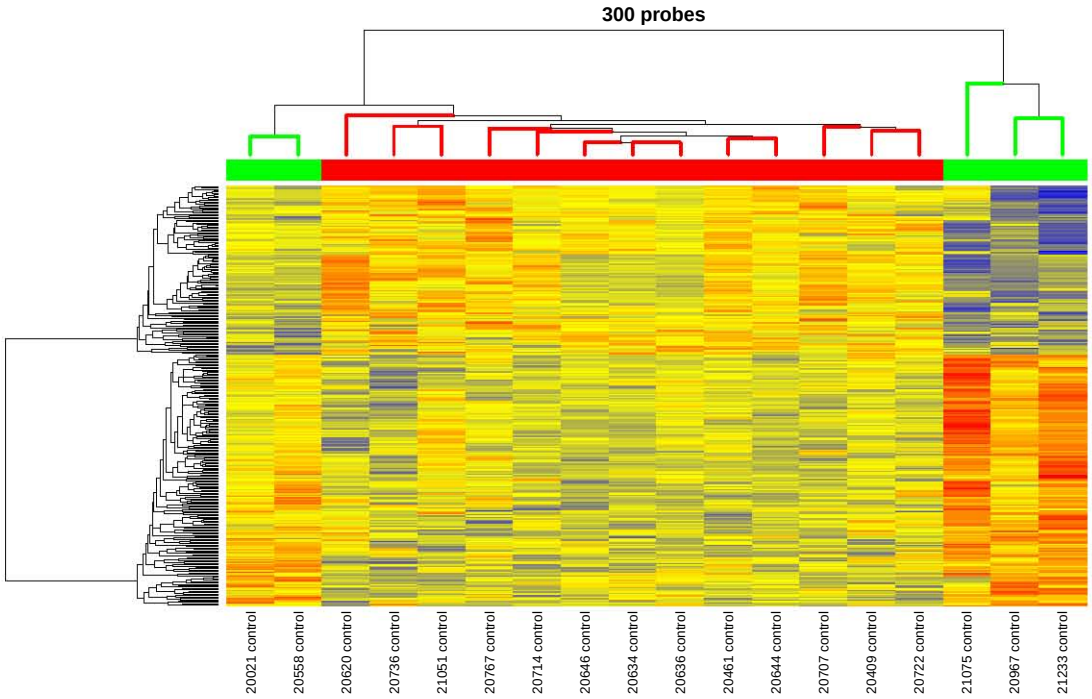
11.1.3 Results for context: not Active TB (datasets: Blood Healthy, Blood LTBI)

Figure 11.1 shows reasonable clustering of the input data from HIV-1 infected and -uninfected samples for the top 300 differentially expressed probes. This shows that there probably is a specific signal of HIV-1 infection, but requires larger numbers to detect reliably. Cell-type specific differential expression was detectable, and shows different results depending on whether prior immune sensitisation to *M. tuberculosis* had occurred (LTBI). In both groups, NK cell and neutrophil sig-

nals are evident, based on the FDR plots shown in Figure 11.2. In addition, "antigen presenting cells" (comprising monocytes and dendritic cells), show a significant FDR drop in HIV-1 uninfected samples, while lymphocytes show possible signal in HIV-1 infected individuals. Please note that due to small sample numbers in this analysis we show cell-type specific responses for groups of cells. Module detection was similar in both groups as shown by the module detection plots in Figure 11.3. Surprisingly, very few probes of either the top 300 or top 500 differentially expressed probes in either set overlap. A number of possible explanations for exist, but remain speculative given the high probability of false positives in the probes shown.

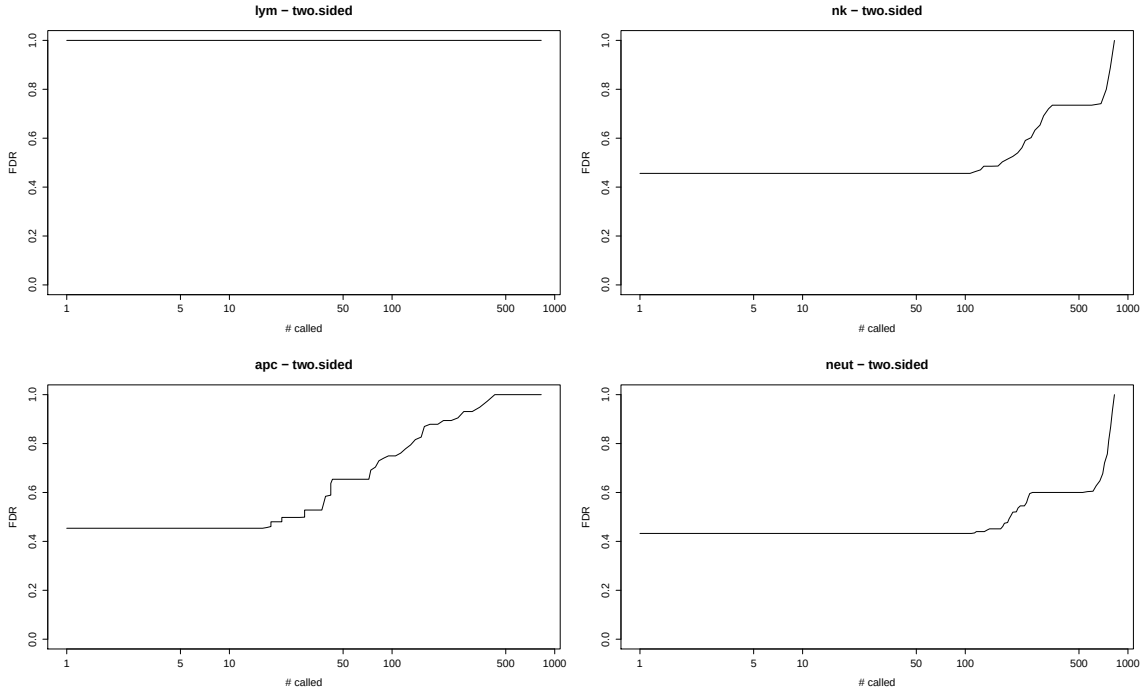


(a) **Heatmap of differentially expressed probes, Healthy**

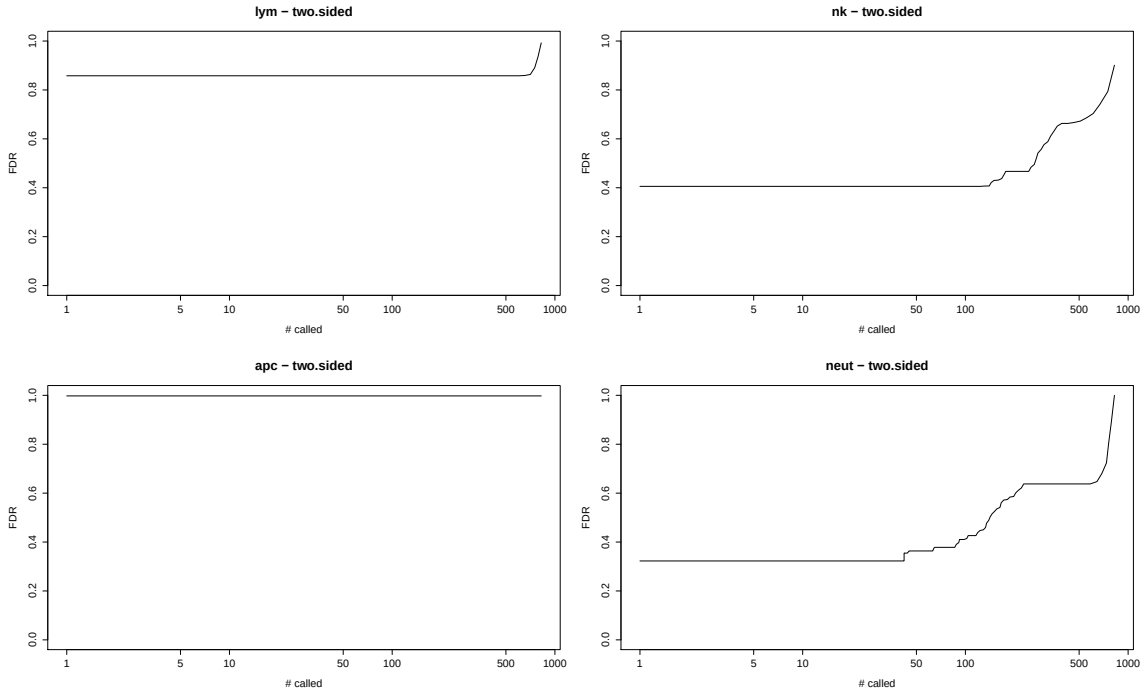


(b) **Heatmap of differentially expressed probes, LTBI**

Figure 11.1: **Results for contrast *HIV status* in contexts *Healthy* and *LTBI*.** Values are row-scaled, and range from low (blue) to intermediate (yellow) to high (red). Samples are clustered using Spearman rank correlation, and probes are clustered using Pearson correlation. Sample colours: green: *HIV negative*, red: *HIV positive*. Plot titles reflect the number of included probes.

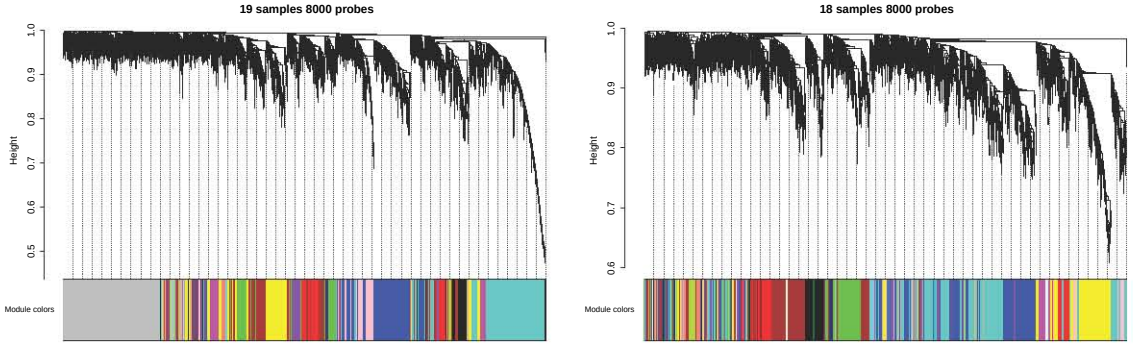


(a) **False discovery rate plot, healthy**



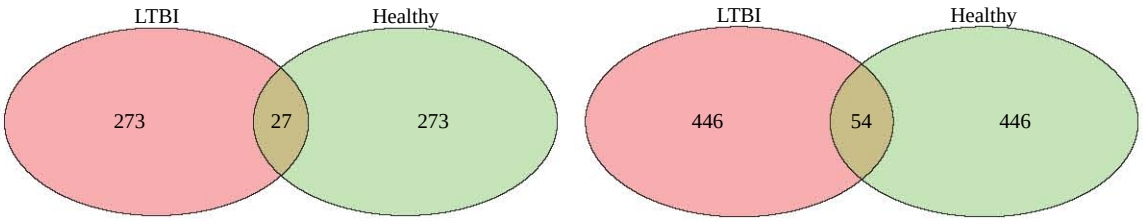
(b) **False discovery rate plot, LTBI**

Figure 11.2: **Cell-type specific differential expression.**



(a) **Module clustering Healthy** (b) **Module clustering LTBI**

Figure 11.3: **Module clustering.**



(a) **Overlap of significantly differentially expressed probes** (b) **Overlap of top 500 differentially expressed probes**

Figure 11.4: **Overlap of differentially expressed probes.**

11.2 Question 6: HIV (active TB)

11.2.1 Included patients and samples

This subsection addresses HIV-1 and *Mycobacterium tuberculosis* co-infection. Four datasets were included in this analysis: *"blood, active TB*", *"pericardial fluid, active TB-PC*", *"blood, PTB"* and *"blood, active TB-PC*". Table 11.3 shows the clinical characteristics of the included study subjects. Between datasets, the variables sex (HIV-positive) and CD4 count (HIV-positive) were not matched. Within the two datasets for which results are presented in this thesis (*"pericardial fluid, active TB-PC*" and *"blood, active TB-PC*", the variables age and CD4 count are not matched. The HIV-1 infected groups were younger and had lower CD4 counts, in keeping with a higher risk for extrapulmonary TB at a younger age driven by lower CD4 counts.

11.2.2 Overall results

The overall results (Table 11.4) show similar numbers of differentially expressed probes prior to multiple testing correction, with no probes significant after multiple testing. This is likely due to the small number of included samples, and the results should be treated with reserve. The top 300 differentially expressed probes based on log-likelihood of differential expression were used for subsequent analysis.

11.2.3 Results for context: Compartment (datasets: Blood TB-PC, Fluid TB-PC)

Figure 11.5 shows reasonable clustering of the input data from HIV-1 infected and -uninfected samples for the top 300 differentially expressed probes in both blood and pericardial fluid. In both cases, the HIV-1 uninfected individuals are assigned to a subcluster of HIV-1 infected individuals. This shows that there possibly is a specific signal of HIV-1 infection, but requires larger numbers to detect reliably, and this signal may be influenced by active tuberculosis. Cell-type specific differential expression was detectable, and shows different results depending on compartment. In blood, the strongest signal is found in neutrophils, while in pericardial fluid all four groups of cells show evidence of cell-specific differential expression (see Figure 11.6). Again, due to small sample num-

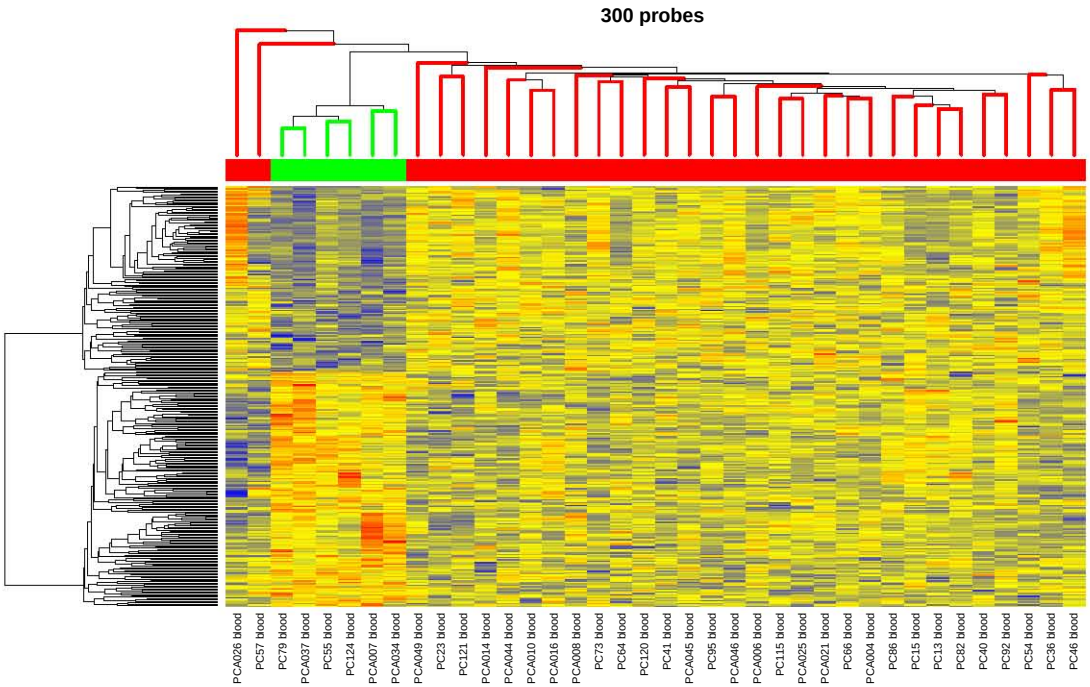
Table 11.3: **Clinical characteristics: HIV positive vs HIV negative (active TB).** Two-way comparison of demographic variables and CD4 count between and with groups of study participants. P-values less than 0.05 are regarded as significant.

	Blood activeTB HIVpos-HIVneg			Fluid activeTB HIVpos-HIVneg			Blood PTB HIVpos-HIVneg			Blood TB-PC HIVpos-HIVneg			Pval	Test
Age	Median	LQ	UQ	Median	LQ	UQ	Median	LQ	UQ	Median	LQ	UQ	Pval	Test
age_All	33.73	29.84	42.23	33.13	29.6	38.51	33.85	28.12	42.27	33.47	30.17	41.39	0.966	Kruskal-Wallis rank
age_ negative	40.67	33.69	52.86	53.32	35.95	56.07	33.85	31.78	38.6	46.6	38.31	54	0.425	Kruskal-Wallis rank
age_ positive	33.3	29.6	38.6	31.5	28.51	37.53	34.25	28.12	42.27	33.03	29.7	38.06	0.806	Kruskal-Wallis rank
age_P value	0.065			0.015			0.888			0.010				
age_Test	Kruskal-Wallis rank			Kruskal-Wallis rank			Kruskal-Wallis rank			Kruskal-Wallis rank				
Sex	ratio F/M	Female	Male	ratio F/M	Female	Male	ratio F/M	Female	Male	ratio F/M	Female	Male	Pval	test
sex_All	1.148	31	27	0.765	13	17	3.667	11	3	1.000	19	19	0.177	Fisher's Exact Test
sex_ negative	0.714	5	7	1.500	3	2	0.333	1	3	1.000	3	3	0.814	Fisher's Exact Test
sex_ positive	1.300	26	20	0.667	10	15	Inf	10	0	1.000	16	16	0.007	Fisher's Exact Test
sex_P value	0.553			0.742			0.018			1.000				
sex_Test	2-sample test for eq			2-sample test for eq			2-sample test for eq			2-sample test for eq				
CD4	Median	LQ	UQ	Median	LQ	UQ	Median	LQ	UQ	Median	LQ	UQ	Pval	Test
CD4_All	210	92	324.5	111	86	273	322	267.5	454.5	125	86	277.5	0.016	Kruskal-Wallis rank
CD4_ negative	566	404.5	653	607.5	492.8	651				566	282	649	NA	NA
CD4_ positive	167	86	291.5	105	81	184	322	267.5	454.5	110	82.25	214.5	0.001	Kruskal-Wallis rank
CD4_P value	0.001			0.004			NA			0.002				
CD4_Test	Kruskal-Wallis rank			Kruskal-Wallis rank			NA			Kruskal-Wallis rank				

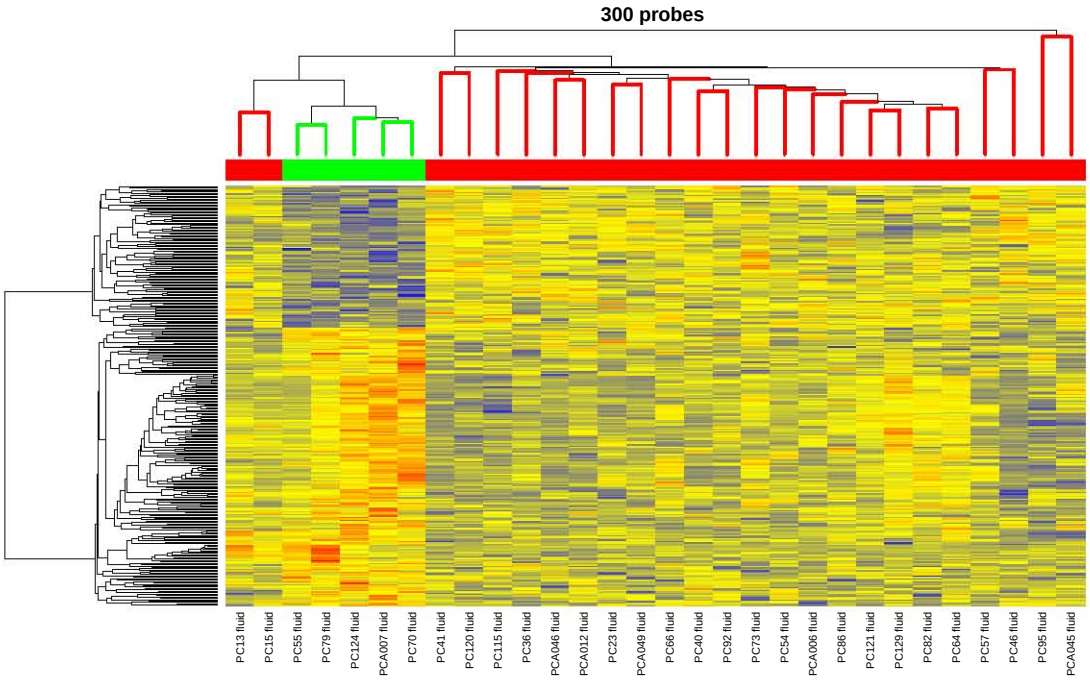
Table 11.4: **Overall results: HIV positive vs HIV negative (active TB).** Output of the analysis script, showing the numbers of included samples, and statistics for differential expression, deconvolution and gene co-expression network analysis. P-values less than 0.05 are regarded as significant.

Question 6: HIV (active TB)		
Contrast: HIV neg vs. HIV pos		
Dataset	1	2
Context	Blood, TB-PC	Fluid, TB-PC
N: HIV neg	6	5
N: HIV pos	32	25
Analysis 1: Differential expression		
N significant probes (unadjusted)	2098	1778
N significant probes (BH adjusted)	0	0
N probes used (heatmaps, csDE)	300	300
Analysis 2: Deconvolution		
Deconvolution successful	yes	yes
N of N significant for contrast (BH)	0 of 13	1 of 13
N of N PBMC significant for contrast (BH)	0 of 12	1 of 12
Cell-specific DE successful	yes	yes
Number of cell types	4	4
N cell types that reach FDR < 0.4	0	2
Analysis 3: WGCNA		
R^2 cutoff for scale-free approximation	0.8	0.8
Soft threshold power	6	7
Number of modules	20	16
Probes assigned to modules (N, %)	4768 (60)	4425 (55)
GO: Entrez IDs submitted	4100	4014
GO: Entrez IDs mapped	2698	2653
N (%) modules cor GS/MM sig	6 (30)	8 (50)
N (%) modules sig for contrast (BH)	19 (95)	12 (75)

bers in this analysis we show cell-type specific responses for groups of cells. Module detection was similar in both groups as shown by the module detection plots in Figure 11.7. A large number of probes were not assigned to modules. Very few probes of either the top 300 or top 500 differentially expressed probes in either set overlap. This is not surprising, as the site of disease has been shown in previous work to cause changes in HIV-1 viral dynamics relative to blood [49].

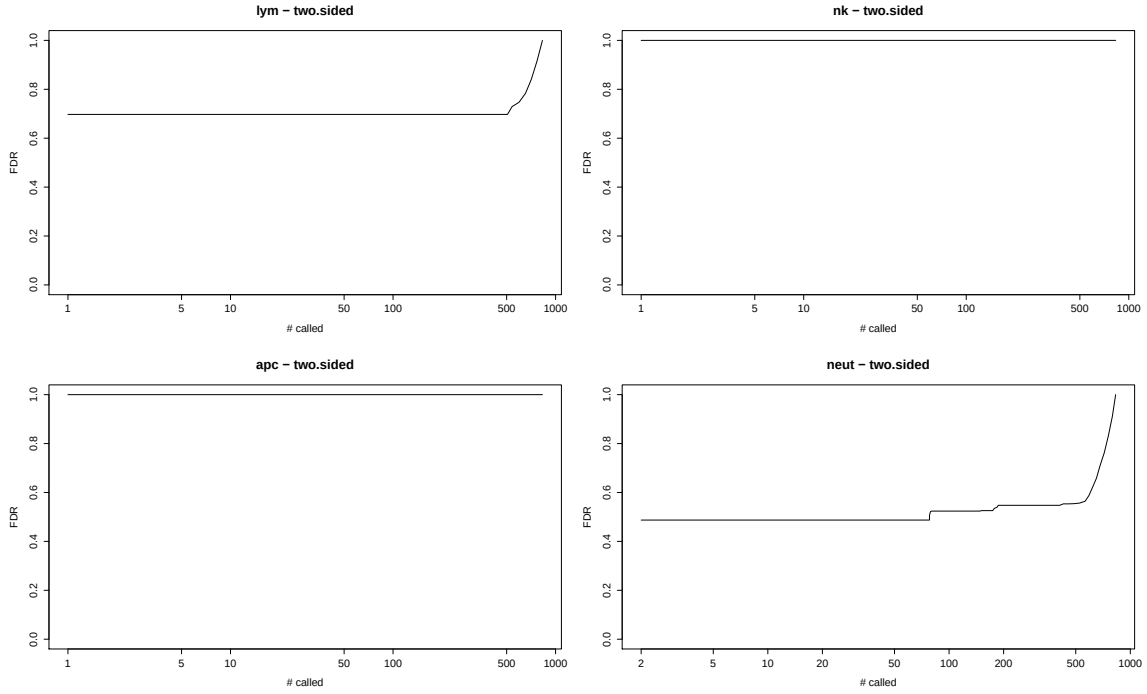


(a) **Heatmap of differentially expressed probes, blood**

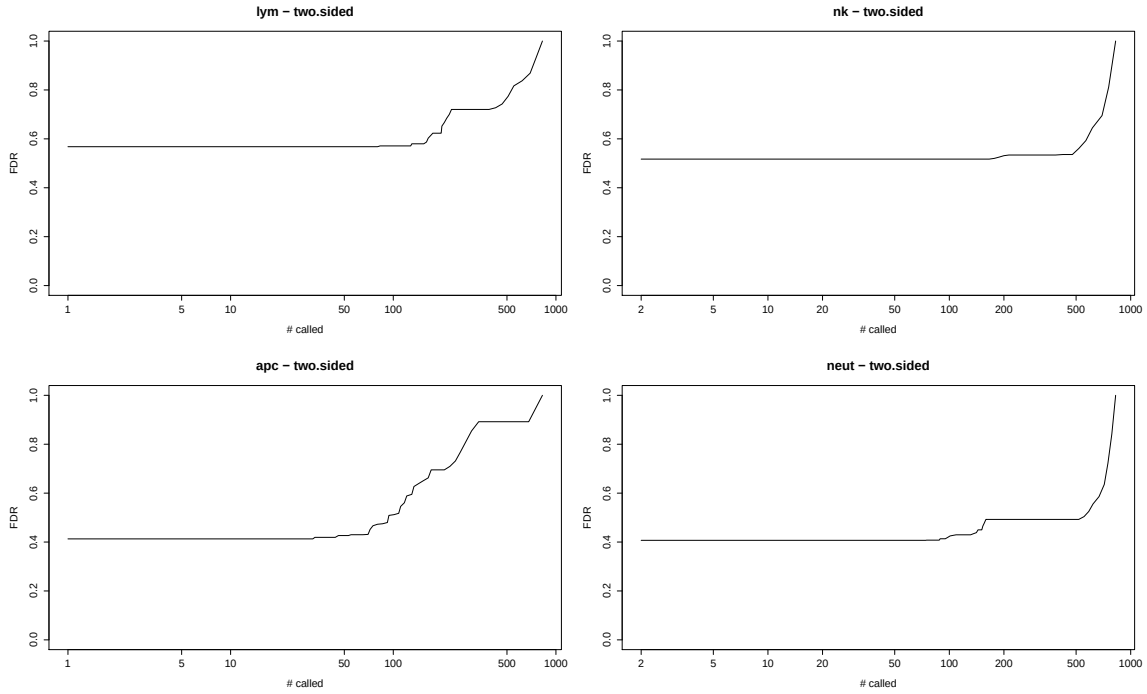


(b) **Heatmap of differentially expressed probes, fluid**

Figure 11.5: **Results for contrast *HIV status* in contexts *TB-PC Blood* and *TB-PC Fluid.*** Values are row-scaled, and range from low (blue) to intermediate (yellow) to high (red). Samples are clustered using Spearman rank correlation, and probes are clustered using Pearson correlation. Sample colours: green: *HIV negative*, red: *HIV positive.* Plot titles reflect the number of included probes.

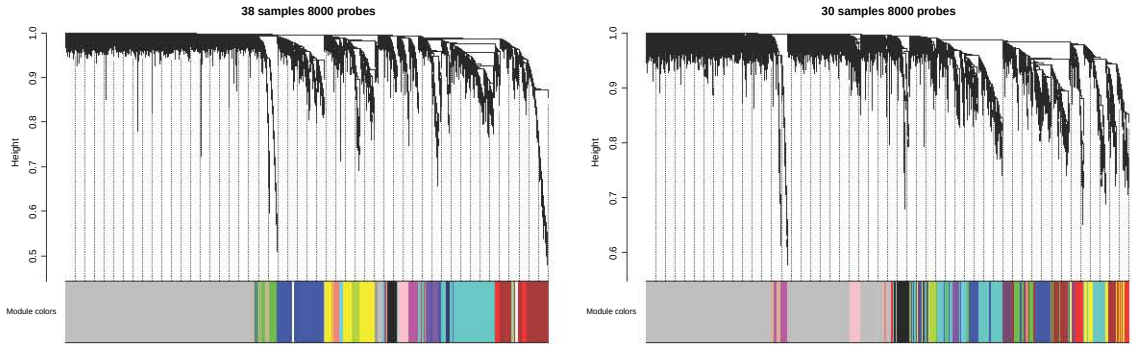


(a) **False discovery rate plot blood**



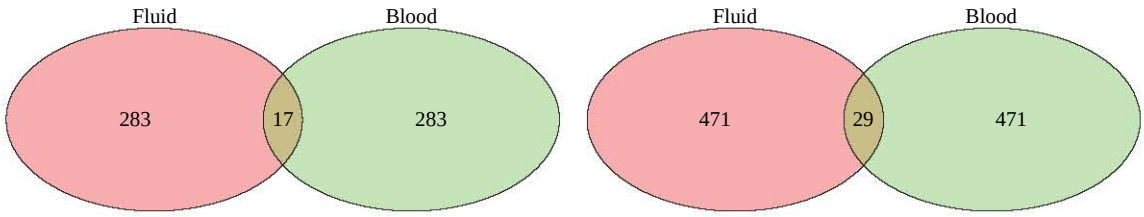
(b) **False discovery rate plot pericardial fluid**

Figure 11.6: **Cell-type specific differential expression.**



(a) **Module clustering Blood** (b) **Module clustering Fluid**

Figure 11.7: **Module clustering.**



(a) **Overlap of significantly differentially expressed probes** (b) **Overlap of top 500 differentially expressed probes**

Figure 11.8: **Overlap of differentially expressed probes.**

11.3 Pathway analysis

Pathway analysis was performed as described in Section 3.8.8. Tables 11.5, 11.6, 11.7 and 11.8 list the pathways found to be significantly differentially regulated in HIV-1 infection in two groups: *not active TB* and *active TB* (see Sections 11.1 and 11.2) . Figure 11.9 shows heatmaps for the differentially regulated pathways in HIV-1 infection.

Table 11.5: **Upregulated pathways in HIV (not active TB).** Colours: red = also upregulated in HIV-1 infection in the active TB group, yellow = downregulated in HIV-1 infection in the active TB group

Pathway	Mean logFC
hsa03050 Proteasome	1.490
hsa04612 Antigen processing and presentation	1.305
hsa00190 Oxidative phosphorylation	1.212
hsa04672 Intestinal immune network for IgA production	0.868
hsa04623 Cytosolic DNA-sensing pathway	0.853
hsa03040 Spliceosome	0.790
hsa04622 RIG-I-like receptor signaling pathway	0.777
hsa03030 DNA replication	0.756
hsa04145 Phagosome	0.743
hsa04514 Cell adhesion molecules (CAMs)	0.727
hsa00970 Aminoacyl-tRNA biosynthesis	0.677
hsa00010 Glycolysis / Gluconeogenesis	0.662
hsa00240 Pyrimidine metabolism	0.633
hsa04110 Cell cycle	0.623
hsa03008 Ribosome biogenesis in eukaryotes	0.624
hsa00020 Citrate cycle (TCA cycle)	0.602
hsa04260 Cardiac muscle contraction	0.543
hsa03060 Protein export	0.538
hsa00051 Fructose and mannose metabolism	0.489
hsa00100 Steroid biosynthesis	0.485
hsa00480 Glutathione metabolism	0.456
hsa00290 Valine, leucine and isoleucine biosynthesis	0.462
hsa00360 Phenylalanine metabolism	0.434

Table 11.6: **Downregulated pathways in HIV (not active TB).** Colours: green = also downregulated in HIV-1 infection in the active TB group, yellow = upregulated in HIV-1 infection in the active TB group

Pathway	Mean logFC
hsa04012 ErbB signaling pathway	-0.824
hsa04910 Insulin signaling pathway	-0.751
hsa04150 mTOR signaling pathway	-0.749
hsa04144 Endocytosis	-0.729
hsa04010 MAPK signaling pathway	-0.693
hsa04360 Axon guidance	-0.681
hsa04960 Aldosterone-regulated sodium reabsorption	-0.680
hsa00590 Arachidonic acid metabolism	-0.669
hsa03010 Ribosome	-0.694
hsa04630 Jak-STAT signaling pathway	-0.617
hsa04320 Dorso-ventral axis formation	-0.610
hsa04310 Wnt signaling pathway	-0.600
hsa04720 Long-term potentiation	-0.583
hsa04916 Melanogenesis	-0.557
hsa04722 Neurotrophin signaling pathway	-0.555
hsa04510 Focal adhesion	-0.538
hsa04666 Fc gamma R-mediated phagocytosis	-0.535
hsa04664 Fc epsilon RI signaling pathway	-0.506
hsa04070 Phosphatidylinositol signaling system	-0.505
hsa04710 Circadian rhythm - mammal	-0.507
hsa00601 Glycosphingolipid biosynthesis - lacto and neolacto series	-0.507
hsa04350 TGF-beta signaling pathway	-0.500
hsa04340 Hedgehog signaling pathway	-0.500
hsa04210 Apoptosis	-0.500
hsa04912 GnRH signaling pathway	-0.463
hsa04810 Regulation of actin cytoskeleton	-0.454
hsa04730 Long-term depression	-0.445
hsa04920 Adipocytokine signaling pathway	-0.443
hsa04370 VEGF signaling pathway	-0.430
hsa00562 Inositol phosphate metabolism	-0.430
hsa04740 Olfactory transduction	-0.422
hsa04330 Notch signaling pathway	-0.409
hsa04966 Collecting duct acid secretion	-0.409

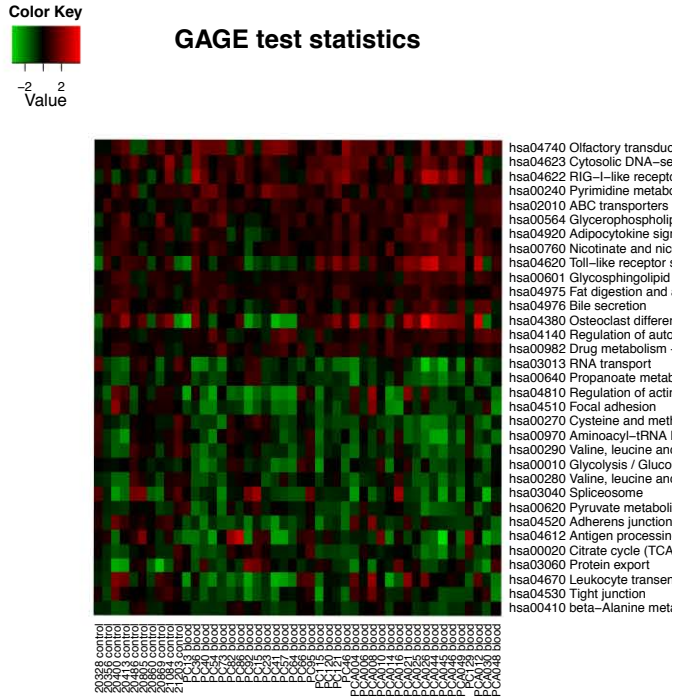
Table 11.7: **Upregulated pathways in HIV (active TB).** Colours: red = also upregulated in HIV-1 infection in the not active TB group, yellow = downregulated in HIV-1 infection in the not active TB group

Pathway	Mean logFC
hsa04740 Olfactory transduction	0.769
hsa04623 Cytosolic DNA-sensing pathway	0.742
hsa04622 RIG-I-like receptor signaling pathway	0.714
hsa00240 Pyrimidine metabolism	0.623
hsa02010 ABC transporters	0.569
hsa00564 Glycerophospholipid metabolism	0.539
hsa04920 Adipocytokine signaling pathway	0.485
hsa00760 Nicotinate and nicotinamide metabolism	0.488
hsa04620 Toll-like receptor signaling pathway	0.458
hsa00601 Glycosphingolipid biosynthesis - lacto and neolacto series	0.439
hsa04975 Fat digestion and absorption	0.433
hsa04976 Bile secretion	0.396
hsa04380 Osteoclast differentiation	0.382
hsa04140 Regulation of autophagy	0.377
hsa00982 Drug metabolism - cytochrome P450	0.359

Table 11.8: **Downregulated pathways in HIV (active TB).** Colours: green = also downregulated in HIV-1 infection in the not active TB group, yellow = upregulated in HIV-1 infection in the not active TB group

Pathway	Mean logFC
hsa03013 RNA transport	-0.735
hsa00640 Propanoate metabolism	-0.581
hsa04810 Regulation of actin cytoskeleton	-0.560
hsa04510 Focal adhesion	-0.546
hsa00270 Cysteine and methionine metabolism	-0.531
hsa00970 Aminoacyl-tRNA biosynthesis	-0.532
hsa00290 Valine, leucine and isoleucine biosynthesis	-0.522
hsa00010 Glycolysis / Gluconeogenesis	-0.473
hsa00280 Valine, leucine and isoleucine degradation	-0.471
hsa03040 Spliceosome	-0.455
hsa00620 Pyruvate metabolism	-0.438
hsa04520 Adherens junction	-0.425
hsa04612 Antigen processing and presentation	-0.421
hsa00020 Citrate cycle (TCA cycle)	-0.408
hsa03060 Protein export	-0.394
hsa04670 Leukocyte transendothelial migration	-0.362
hsa04530 Tight junction	-0.354
hsa00410 beta-Alanine metabolism	-0.352

(a) **Not active TB**



(b) **Active TB**

Figure 11.9: **Significant pathways in HIV.** The heatmaps show mean estimates for pathway expression for all HIV-1 infected samples (HIV-1 uninfected samples are not shown). Colours: *red* = pathway upregulated; *green* = pathway downregulated.

Figure 11.10 shows the extent to which pathways up- or downregulated by HIV-1 infection are shared by individuals with and without active tuberculosis. This provides additional information about possible disease mechanisms in HIV-TB. In contrast to the shared pathways affected by active tuberculosis, pathways affected by HIV show very little overlap in the two groups (not active TB and active TB). This suggests that active tuberculosis is an important co-regulator of HIV-1-related pathogenesis. In fact, more pathways are up- or downregulated in *opposite* directions depending on TB status than in the same direction. This is discussed further below.

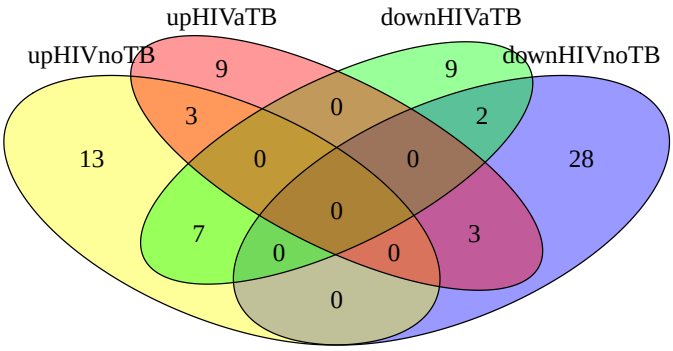


Figure 11.10: **Overlap of differentially regulated pathways in active tuberculosis stratified by HIV status.**

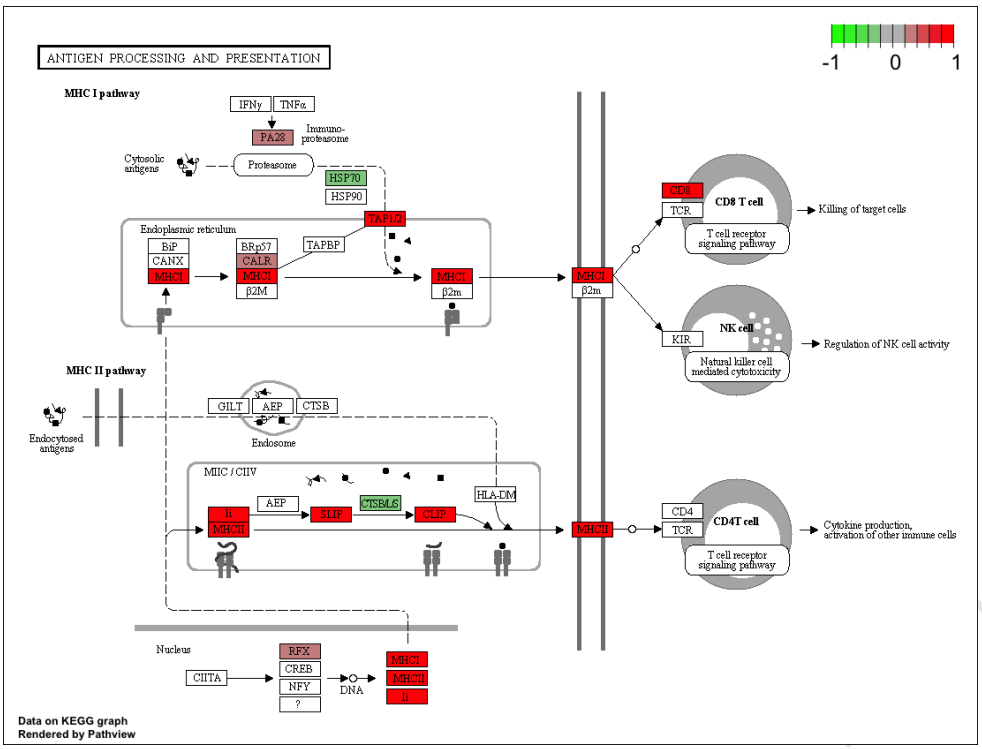
11.3.1 Pathway visualisation

The antigen processing and presentation pathway which is strongly upregulated by HIV-1 in the absence of active tuberculosis is downregulated in HIV-TB co-infection. Figure 11.11 shows the extent of this differential regulation at gene level. This should be compared to Figure 10.21 which demonstrates the converse situation, i.e. the effect of active tuberculosis on the same pathway, with and without HIV-1 co-infection.

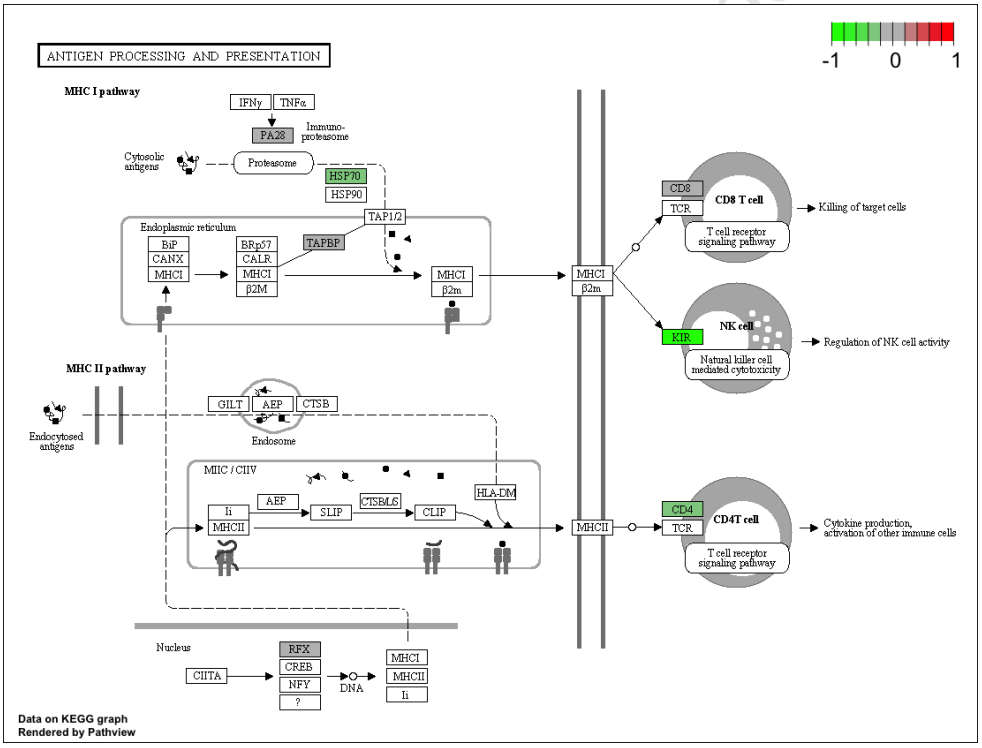
Another interesting pathway concerns innate defences against viral infections mediated by the RIG-1-like pathway (*hsa04622*). This pathway is an important sensor for cytoplasmic viral RNA, essential for generating a downstream signal that results in type I interferon secretion. RIG-1 is induced by type 1 interferon signalling as well as double-stranded viral RNA. While purified HIV-1 RNA indeed produces a RIG-1 dependent interferon response, actual HIV-1 infection does not result in increased interferon production *in vitro*. This was shown to be due to inhibition of

the RIG-1 sensor (a DExD/H box RNA helicase) by HIV-1 protease by reducing phosphorylation of IFN regulatory factor 3 (IRF-3) phosphorylation [162]. The proximal pathway appears to be upregulated during HIV-1 infection in both *active TB* and *not active TB* contexts (see Figure 11.12), but IRF-3, however, is not upregulated. A possible interpretation of the pathway result is that failure of upregulating IRF-3 may impair the type I interferon response, which would relate to a previously published result, in which a similar picture of downregulation of IRF-3 activity, but not IRF-7 was reported *in vitro*. [163].

In summary, pathway analysis with mapping of differentially expressed genes onto differentially regulated pathways provides substantial insight into the complex relationships between HIV-1 and *M. tuberculosis* infections.

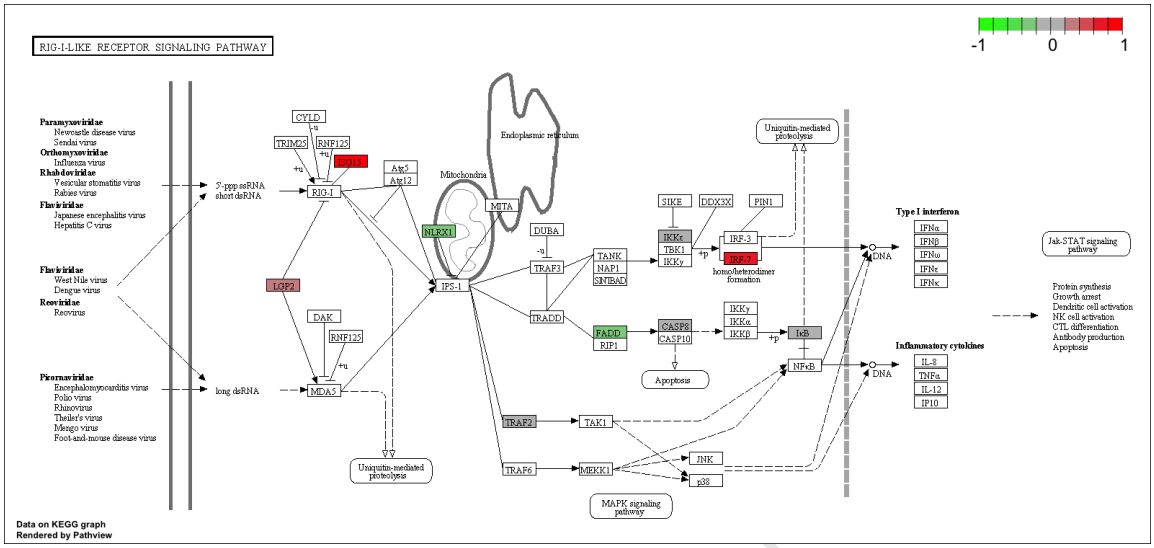


(a) **Not active TB**

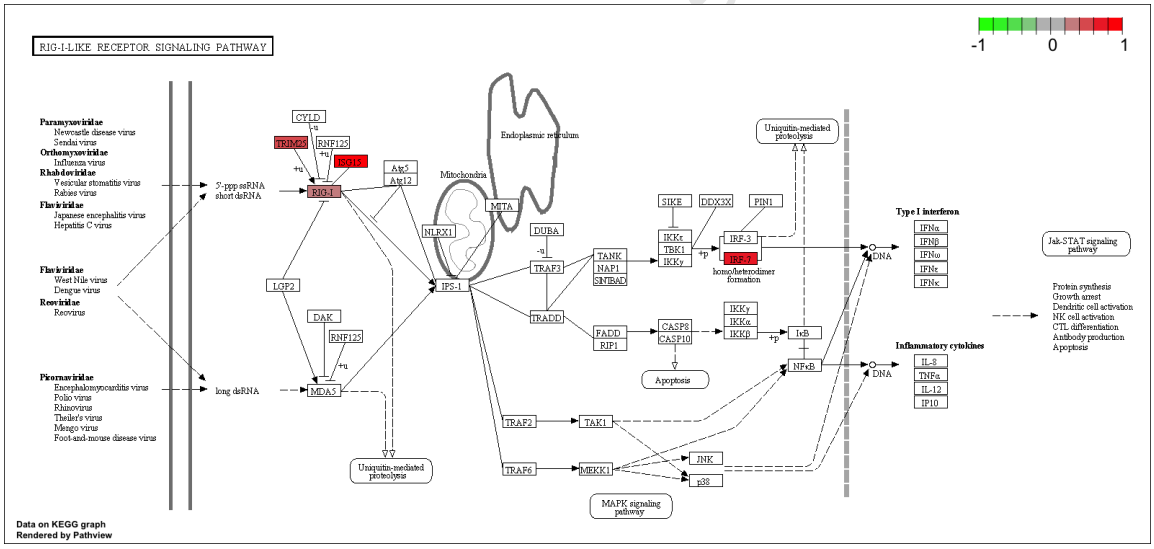


(b) **Active TB**

Figure 11.11: **Antigen processing and presentation pathway.** Canonical pathway *hsa04612* from KEGG, rendered by *pathview* software. The logfold change estimates output by limma are overlaid, normalised to a range -1 to +1. Colours: *red* = upregulated in HIV-1 infection; *green* = downregulated in HIV-1 infection.



(a) **Not active TB**



(b) **Active TB**

Figure 11.12: **RIG-I-like receptor signaling pathway.** Canonical pathway *hsa04622* from KEGG, rendered by *pathview* software. The logfold change estimates output by limma are overlaid, normalised to a range -1 to +1. Colours: *red* = upregulated in HIV-1 infection; *green* = downregulated in HIV-1 infection.

12 Contrasts involving the compartment

Chapter overview

In this Chapter I discuss the results of transcriptomic analysis at the site of active tuberculosis in individuals with tuberculous pericarditis.

12.1 Question 7: Blood and pericardial fluid in TB-PC

While blood transcriptomic responses are informative regarding the nature of the immune response to tuberculosis with and without HIV-1 co-infection, the site of disease is of considerable interest in studies of tuberculosis pathogenesis, as it is at the site of infection that the disease develops and progresses. Given our understanding of transcriptional responses detectable in blood, it is instructive to assess transcriptional responses in pericardial fluid relative to those in blood.

12.1.1 Included patients and samples

Five datasets were included in this analysis: all matched blood/pericardial fluid samples, and matched blood pericardial fluid samples for HIV-1 uninfected, HIV-1 infected, Effusive phenotype and effusive-constrictive phenotype subgroups[1]. All five sets/ subsets were matched for age and sex; the HIV-1 infected subsets had significantly lower CD4 counts. Results for two contexts will be reported here (TB-PC HIV negative, TB-PC HIV positive). As only matched blood and pericardial fluid samples were included, there is no bias due to mismatched demographic or phenotypic variables, setting this analysis apart from all others.

[1]The much smaller sample sizes in the haemodynamic phenotype subsets (*EFF blood - fluid* and *EC blood - fluid*) are due to the limited number of study subjects included in this study who had right heart studies at baseline.

Table 12.1: **Clinical characteristics: blood vs pericardial fluid.** Two-way comparison of demographic variables and CD4 count between and with groups of study participants. P-values less than 0.05 are regarded as significant. This table includes data for the combined group ("not active TB"), LTBI and healthy.

	TB-PC all blood-fluid			TB-PC HIVneg blood-fluid			TB-PC HIVpos blood-fluid			TB-PC HIVpos EFF blood-fluid			TB-PC HIVpos EC blood-fluid			Pval	Test
Age	Median	LQ	UQ	Median	LQ	UQ	Median	LQ	UQ	Median	LQ	UQ	Median	LQ	UQ	Pval	Test
age_All	33.08	29.6	38.7	34.66	30.24	56.07	32.98	29.54	37.83	30.12	27.12	34.87	33.08	32.92	38.75	0.143	Kruskal-Wallis rank
age_ blood	33.08	29.65	38.67	34.66	30.79	55.38	32.98	29.54	37.83	30.12	27.51	33.7	33.08	32.92	38.75	0.491	Kruskal-Wallis rank
age_ fluid	33.08	29.65	38.67	34.66	30.79	55.38	32.98	29.54	37.83	30.12	27.51	33.7	33.08	32.92	38.75	0.491	Kruskal-Wallis rank
age_P value	1.000			1.000			1.000			1.000			1.000				
age_Test	Kruskal-Wallis rank			Kruskal-Wallis rank			Kruskal-Wallis rank			Kruskal-Wallis rank			Kruskal-Wallis rank				
Sex	ratio F/M	Female	Male	ratio F/M	Female	Male	ratio F/M	Female	Male	ratio F/M	Female	Male	ratio F/M	Female	Male	Pval	test
sex_All	0.645	40	62	1.000	10	10	0.577	30	52	0.833	10	12	0.800	8	10	0.783	Fisher's Exact Test
sex_ blood	0.645	20	31	1.000	5	5	0.577	15	26	0.833	5	6	0.800	4	5	0.923	Fisher's Exact Test
sex_ fluid	0.645	20	31	1.000	5	5	0.577	15	26	0.833	5	6	0.800	4	5	0.923	Fisher's Exact Test
sex_P value	1.000			1.000			1.000			1.000			1.000				
sex_Test	2-sample test for eq			2-sample test for eq			2-sample test for eq			2-sample test for eq			2-sample test for eq				
CD4	Median	LQ	UQ	Median	LQ	UQ	Median	LQ	UQ	Median	LQ	UQ	Median	LQ	UQ	Pval	Test
CD4_All	182	86	326	566	412	657	118	74	279	86	46	174.2	105	95	223	0.000	Kruskal-Wallis rank
CD4_ blood	182	86	326	566	412	657	118	74	279	86	49	166.5	105	95	223	0.000	Kruskal-Wallis rank
CD4_ fluid	182	86	326	566	412	657	118	74	279	86	49	166.5	105	95	223	0.000	Kruskal-Wallis rank
CD4_P value	1.000			1.000			1.000			1.000			1.000				
CD4_Test	Kruskal-Wallis rank			Kruskal-Wallis rank			Kruskal-Wallis rank			Kruskal-Wallis rank			Kruskal-Wallis rank				

12.1.2 Overall results

Table 12.2 lists the overall results of the three analyses for two datasets. Of all analyses, the contrast "blood vs. pericardial fluid" yielded the highest number of significantly differentially expressed probes. This is likely due to the markedly different cellular composition of pericardial fluid compared to blood (see below). For this reason, cell-type specific differential expression is of great interest here. However, as I will show in the pathway analysis section, global transcriptional changes at the site of disease regardless of cellular origin are still informative about biological processes at the site of disease, which demonstrate a very strong signal indicative of major reprogramming of host metabolism, which is likely to benefit *M. tuberculosis*. Given the significant differences in cellular composition, the modules identified using weighted gene co-expression network analysis possibly represent cell-types. This requires further analysis.

Table 12.2: **Overall results blood vs pericardial fluid.** Output of the analysis script, showing the numbers of included samples, and statistics for differential expression, deconvolution and gene co-expression network analysis. P-values less than 0.05 are regarded as significant.

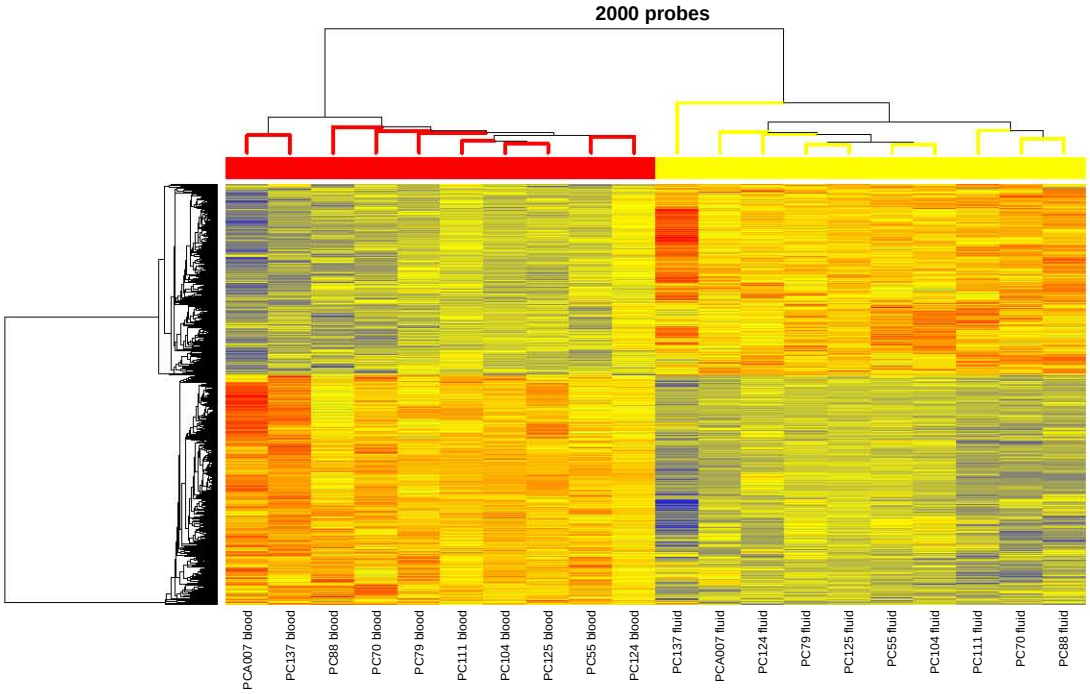
Question 7: Compartment		
Contrast: Blood vs. Pericardial Fluid		
Dataset	1	2
Context	TB-PC HIV neg	TB-PC HIV pos
N: Blood	10	41
N: Fluid	10	41
Analysis 1: Differential expression		
N significant probes (unadjusted)	7895	12688
N significant probes (BH adjusted)	4211	10434
N probes used (heatmaps, csDE)	2000	2000
Analysis 2: Deconvolution		
Deconvolution successful	yes	yes
N of N significant for contrast (BH)	5 of 15	8 of 15
N of N PBMC significant for contrast (BH)	5 of 14	9 of 14
Cell-specific DE successful	yes	yes
Number of cell types	7	7
N cell types that reach FDR < 0.4	1	3
Analysis 3: WGCNA		
R^2 cutoff for scale-free approximation	0.75	0.8
Soft threshold power	8	7
Number of modules	7	13
Probes assigned to modules (N, %)	8000 (100)	7987 (99.84)
GO: Entrez IDs submitted	4665	4875
GO: Entrez IDs mapped	3131	3275
N (%) modules cor GS/MM sig	6	12
N (%) modules sig for contrast (BH)	7 (100)	13 (100)

12.1.3 Results for context: HIV status (datasets: TB-PC HIV neg, TB-PC HIV pos)

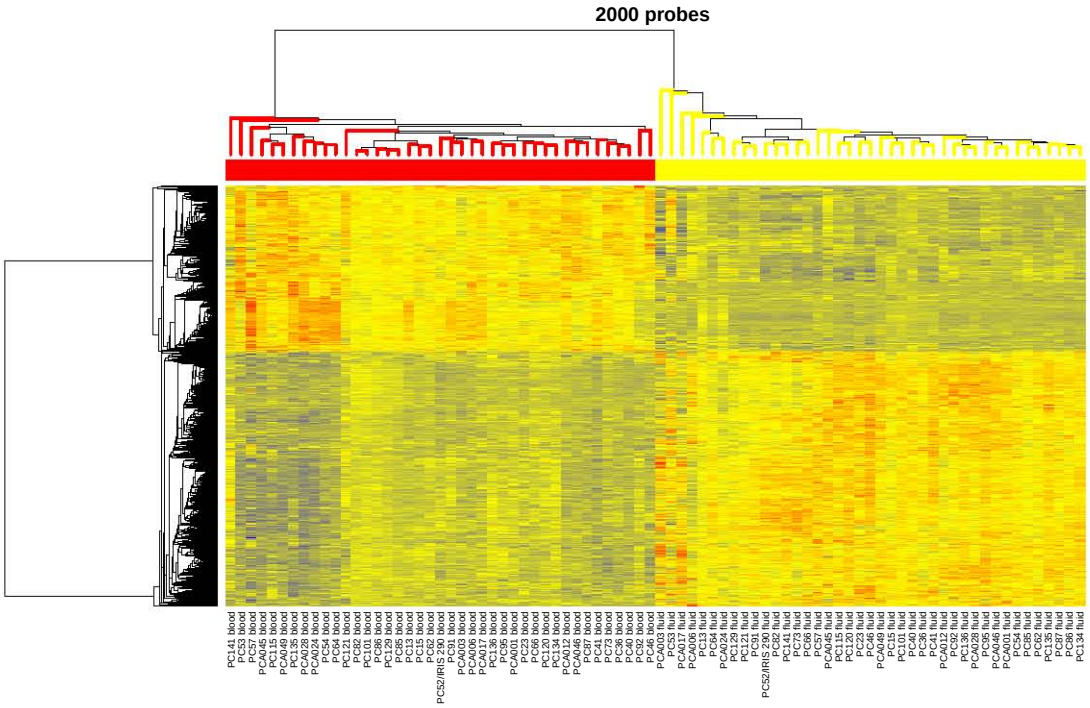
The top 2000 differentially expressed probes in both HIV-1 uninfected and -infected individuals separate the input data into two distinct clusters (blood and pericardial fluid) (Figure 12.1). Of interest is the large overlap between HIV-1 infected and -uninfected individuals when considering the top 2000 differentially expressed probes; in contrast, there is very little overlap between the top 500 differentially expressed probes. Figure 12.2 shows the barplots for proportions of detected cell types. The most striking difference is the low proportion of neutrophils in pericardial fluid. This is in keeping with the most common presentation of tuberculous pericardial effusions as predominantly lymphocytic (see [139] for data in a South African population). As the neutrophils skew the proportions of all other cells (mainly PBMC), barplots of PBMC proportions (normalised to 1.0) are also shown.

Cell-specific differential expression differed between the two groups studied. In HIV-1 uninfected individuals, the strongest signal was detected in B cells, followed by CD8 T cells, with weaker signals in NK cells and dendritic cells. In contrast, HIV-1 infected individuals exhibited strong signals for cell-type specific differential expression in CD4 T cells, B cells, dendritic cells, and to a lesser extent also in NK cells and monocytes.

Virtually all probes examined were assigned to modules. These require further analysis to characterise and relate to functional classes like cell types and pathways.

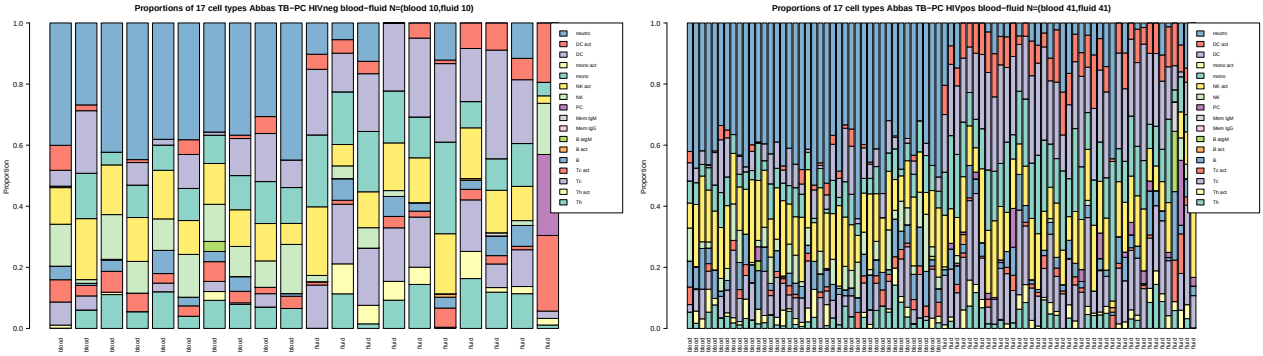


(a) **Heatmap of differentially expressed probes, HIV-1 uninfected**



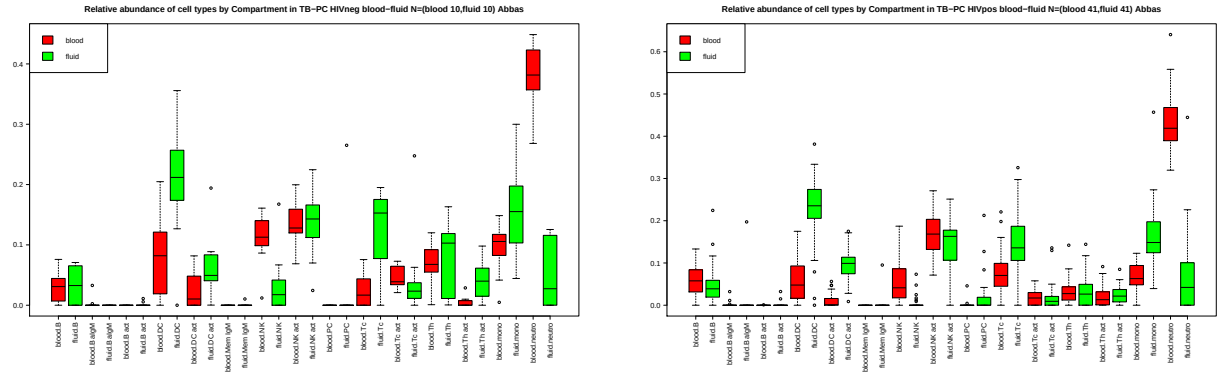
(b) **Heatmap of differentially expressed probes, HIV-1 infected**

Figure 12.1: **Results for contrast *Compartment* in contexts *TB-PC HIV neg* and *TB-PC HIV pos*.** Values are row-scaled, and range from low (blue) to intermediate (yellow) to high (red). Samples are clustered using Spearman rank correlation, and probes are clustered using Pearson correlation. Sample colours: red: *blood*, yellow: *pericardial fluid*. Plot titles reflect the number of included probes.



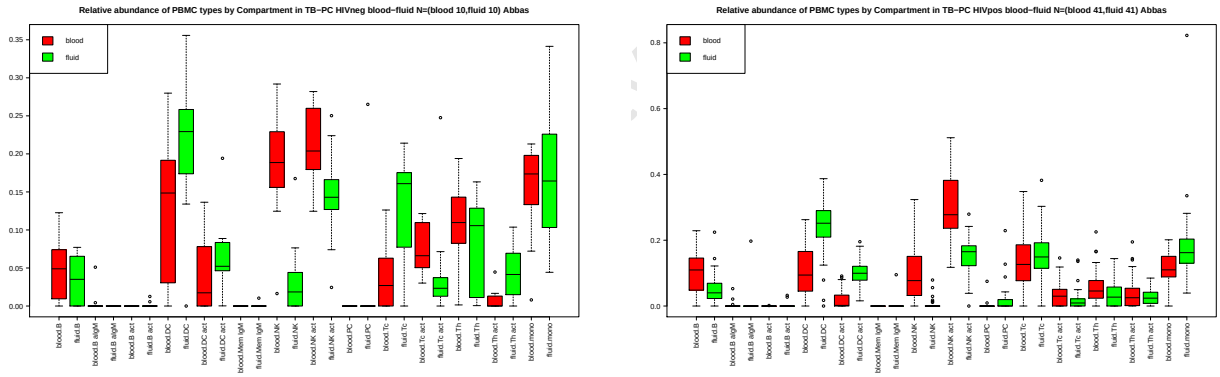
(a) **HIV-1 uninfected: Barplot, all cell types**

(b) **HIV-1 infected: Barplot, all cell types**



(c) **HIV-1 uninfected: Boxplot, all cell types**

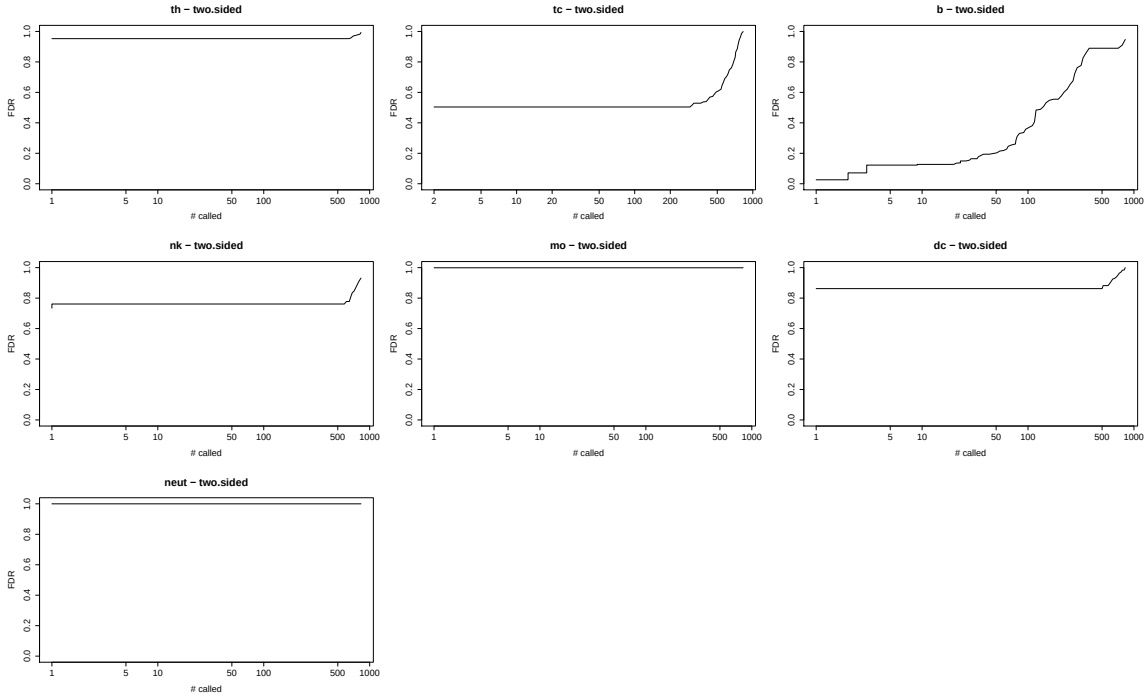
(d) **HIV-1 infected: Boxplot, all cell types**



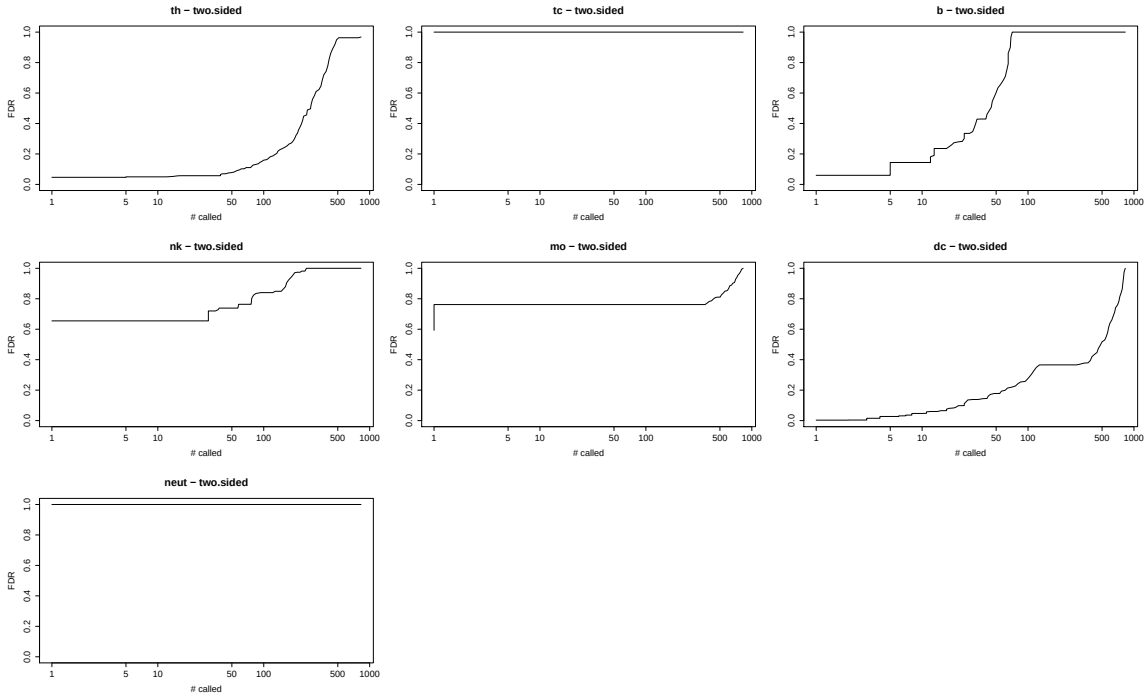
(e) **HIV-1 uninfected: Boxplot, PBMC**

(f) **HIV-1 infected: Boxplot, PBMC**

Figure 12.2: **Barplots and boxplots of matched blood and pericardial fluid samples.** Stacked bar charts for cell proportions for the analysed samples are shown in subfigures a and b. Note the striking visual difference between blood and pericardial fluid. Box-and-whisker charts are shown for all detected cell-types, and for PBMC only in subfigures c, d and e,f, respectively. Each bar is normalised to 1 (100%). Samples are ordered according to disease phenotype (not active TB followed by active TB). The legend lists the cell types identified. Abbreviations: ***neutro***: neutrophils, ***DC act***: activated dendritic cells, ***DC***: non-activated dendritic cells, ***mono***: monocytes, ***NK act***: activated natural killer cells, ***NK***: non-activated natural killer cells, ***B aIGM***: BCR-ligated B cells, ***B***: resting B cells, ***Tc act***: activated CD8 T cell, ***Tc***: resting CD8 T cell, ***Th act***: activated CD4 T cell, ***Th***: resting CD4 T cell.

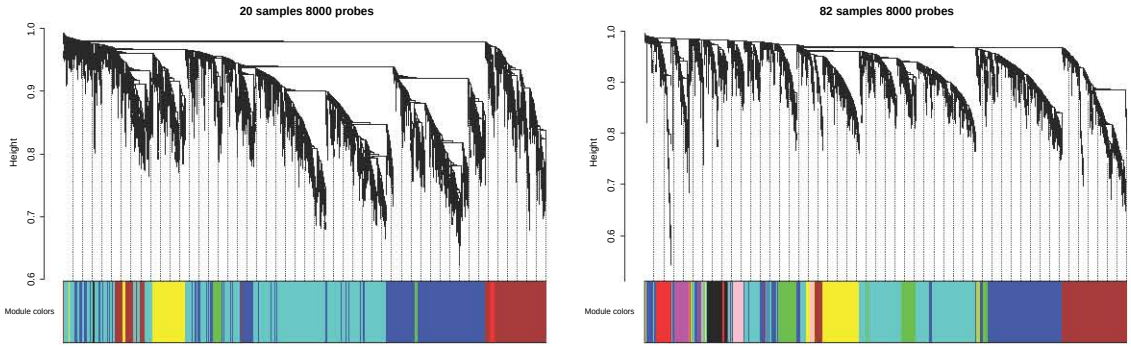


(a) **False discovery rate plot HIV-1 uninfected**



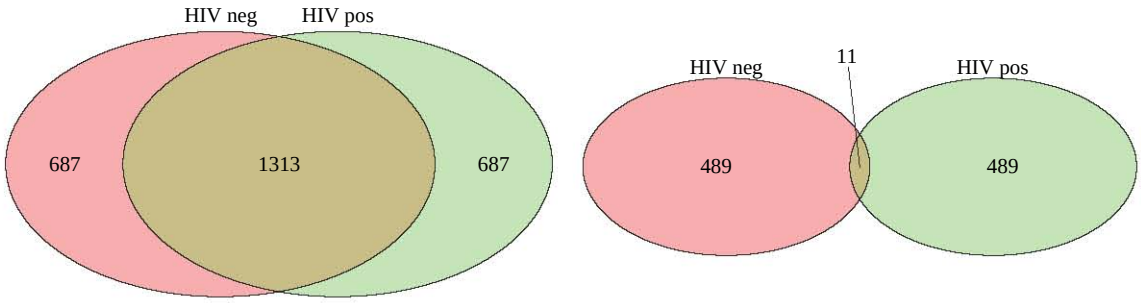
(b) **False discovery rate plot HIV-1 infected**

Figure 12.3: **Cell-type specific differential expression.**



(a) **Module clustering HIV neg** (b) **Module clustering HIV pos**

Figure 12.4: **Module clustering.**



(a) **Overlap of significantly differentially expressed probes** (b) **Overlap of top 500 differentially expressed probes**

Figure 12.5: **Overlap of differentially expressed probes.**

12.2 Pathway analysis

Given the massive differences in differential expression at probe level, pathway analysis is a useful method to reduce the data to more intuitive biological differences. Tables 12.3, 12.4, 12.5 and 12.6 list the pathways up- and downregulated in pericardial fluid (relative to blood) with and without HIV-1 co-infection. Pathway heatmaps are not shown,

Table 12.3: **Pathways significantly upregulated in pericardial fluid (HIV-1 uninfected).** Colours: red = also upregulated in pericardial fluid in the HIV-1 infected group

Pathway	Mean logFC
hsa00970 Aminoacyl-tRNA biosynthesis	2.808
hsa03008 Ribosome biogenesis in eukaryotes	2.587
hsa00010 Glycolysis / Gluconeogenesis	2.136
hsa03040 Spliceosome	2.054
hsa03030 DNA replication	2.106
hsa04142 Lysosome	2.068
hsa03050 Proteasome	1.966
hsa03013 RNA transport	1.896
hsa00020 Citrate cycle (TCA cycle)	1.932
hsa04612 Antigen processing and presentation	1.769
hsa00190 Oxidative phosphorylation	1.724
hsa00290 Valine, leucine and isoleucine biosynthesis	1.657
hsa04672 Intestinal immune network for IgA production	1.495
hsa00310 Lysine degradation	1.488
hsa00280 Valine, leucine and isoleucine degradation	1.364
hsa00640 Propanoate metabolism	1.365
hsa00270 Cysteine and methionine metabolism	1.355
hsa03018 RNA degradation	1.325
hsa00670 One carbon pool by folate	1.296
hsa03060 Protein export	1.308
hsa04141 Protein processing in endoplasmic reticulum	1.188
hsa00051 Fructose and mannose metabolism	1.201
hsa00510 N-Glycan biosynthesis	1.086
hsa00052 Galactose metabolism	1.038
hsa00520 Amino sugar and nucleotide sugar metabolism	0.921
hsa00511 Other glycan degradation	0.932
hsa03410 Base excision repair	0.878
hsa03010 Ribosome	0.889
hsa03020 RNA polymerase	0.820
hsa03430 Mismatch repair	0.814
hsa00620 Pyruvate metabolism	0.773
hsa00630 Glyoxylate and dicarboxylate metabolism	0.752
hsa00410 beta-Alanine metabolism	0.739
hsa00350 Tyrosine metabolism	0.732
hsa04260 Cardiac muscle contraction	0.730
hsa04146 Peroxisome	0.692
hsa00450 Selenocompound metabolism	0.675
hsa00100 Steroid biosynthesis	0.650
hsa00250 Alanine, aspartate and glutamate metabolism	0.638

Table 12.4: **Pathways significantly downregulated in pericardial fluid (HIV-1 uninfected).** Colours: green = also downregulated in pericardial fluid in the HIV-1 infected group

Pathway	Mean logFC
hsa04650 Natural killer cell mediated cytotoxicity	-2.537
hsa04670 Leukocyte transendothelial migration	-2.120
hsa04810 Regulation of actin cytoskeleton	-1.855
hsa04210 Apoptosis	-1.784
hsa04666 Fc gamma R-mediated phagocytosis	-1.607
hsa00564 Glycerophospholipid metabolism	-1.566
hsa04144 Endocytosis	-1.475
hsa04380 Osteoclast differentiation	-1.481
hsa04920 Adipocytokine signaling pathway	-1.338
hsa00910 Nitrogen metabolism	-1.343
hsa04740 Olfactory transduction	-1.312
hsa04966 Collecting duct acid secretion	-1.299
hsa00760 Nicotinate and nicotinamide metabolism	-1.202
hsa04010 MAPK signaling pathway	-1.135
hsa04270 Vascular smooth muscle contraction	-1.121
hsa04140 Regulation of autophagy	-1.118
hsa04664 Fc epsilon RI signaling pathway	-1.105
hsa04720 Long-term potentiation	-1.097
hsa04640 Hematopoietic cell lineage	-1.039
hsa00770 Pantothenate and CoA biosynthesis	-1.007
hsa04062 Chemokine signaling pathway	-0.953
hsa04360 Axon guidance	-0.933
hsa04070 Phosphatidylinositol signaling system	-0.904
hsa00601 Glycosphingolipid biosynthesis - lacto and neolacto series	-0.891
hsa04350 TGF-beta signaling pathway	-0.872
hsa04630 Jak-STAT signaling pathway	-0.838
hsa00562 Inositol phosphate metabolism	-0.837
hsa04730 Long-term depression	-0.824
hsa04614 Renin-angiotensin system	-0.817
hsa04530 Tight junction	-0.804
hsa04510 Focal adhesion	-0.786
hsa04520 Adherens junction	-0.774
hsa04310 Wnt signaling pathway	-0.714
hsa04662 B cell receptor signaling pathway	-0.687
hsa04370 VEGF signaling pathway	-0.657
hsa04115 p53 signaling pathway	-0.649

Table 12.5: **Pathways significantly upregulated in pericardial fluid (HIV-1 infected).** Colours: red = also upregulated in pericardial fluid in the HIV-1 uninfected group

Pathway	Mean logFC
hsa04142 Lysosome	3.400
hsa00970 Aminoacyl-tRNA biosynthesis	2.944
hsa04612 Antigen processing and presentation	2.564
hsa03050 Proteasome	2.532
hsa00010 Glycolysis / Gluconeogenesis	2.300
hsa00190 Oxidative phosphorylation	2.236
hsa03008 Ribosome biogenesis in eukaryotes	2.147
hsa00020 Citrate cycle (TCA cycle)	2.112
hsa04672 Intestinal immune network for IgA production	2.056
hsa03040 Spliceosome	1.704
hsa03030 DNA replication	1.728
hsa03013 RNA transport	1.666
hsa00290 Valine, leucine and isoleucine biosynthesis	1.770
hsa00310 Lysine degradation	1.620
hsa00280 Valine, leucine and isoleucine degradation	1.588
hsa00640 Propanoate metabolism	1.499
hsa00511 Other glycan degradation	1.514
hsa03060 Protein export	1.402
hsa00051 Fructose and mannose metabolism	1.364
hsa00670 One carbon pool by folate	1.333
hsa00520 Amino sugar and nucleotide sugar metabolism	1.273
hsa00270 Cysteine and methionine metabolism	1.225
hsa04141 Protein processing in endoplasmic reticulum	1.143
hsa00052 Galactose metabolism	1.151
hsa00510 N-Glycan biosynthesis	0.983
hsa00410 beta-Alanine metabolism	0.990
hsa00630 Glyoxylate and dicarboxylate metabolism	0.988
hsa04145 Phagosome	0.938
hsa00531 Glycosaminoglycan degradation	0.946
hsa00620 Pyruvate metabolism	0.906
hsa04146 Peroxisome	0.858
hsa04260 Cardiac muscle contraction	0.857
hsa00350 Tyrosine metabolism	0.853
hsa03018 RNA degradation	0.816
hsa03410 Base excision repair	0.810
hsa00480 Glutathione metabolism	0.762
hsa00360 Phenylalanine metabolism	0.751
hsa00071 Fatty acid metabolism	0.694
hsa00250 Alanine, aspartate and glutamate metabolism	0.694
hsa03430 Mismatch repair	0.685
hsa00450 Selenocompound metabolism	0.623
hsa00600 Sphingolipid metabolism	0.602
hsa04610 Complement and coagulation cascades	0.587
hsa00380 Tryptophan metabolism	0.575
hsa00040 Pentose and glucuronate interconversions	0.564
hsa00120 Primary bile acid biosynthesis	0.555
hsa03020 RNA polymerase	0.523
hsa00330 Arginine and proline metabolism	0.509
hsa03320 PPAR signaling pathway	0.508
hsa04621 NOD-like receptor signaling pathway	0.504
hsa00604 Glycosphingolipid biosynthesis - ganglio series	0.503
hsa00603 Glycosphingolipid biosynthesis - globo series	0.486
hsa04973 Carbohydrate digestion and absorption	0.476
hsa00030 Pentose phosphate pathway	0.473
hsa00340 Histidine metabolism	0.448
hsa03440 Homologous recombination	0.437
hsa00650 Butanoate metabolism	0.425
hsa03420 Nucleotide excision repair	0.423
hsa00534 Glycosaminoglycan biosynthesis - heparan sulfate	0.359
hsa04514 Cell adhesion molecules (CAMs)	0.334
hsa04977 Vitamin digestion and absorption	0.281

Table 12.6: **Pathways significantly downregulated in pericardial fluid (HIV-1 infected).** Colours: green = also downregulated in pericardial fluid in the HIV-1 infected group

Pathway	Mean logFC
hsa04650 Natural killer cell mediated cytotoxicity	-2.279
hsa04740 Olfactory transduction	-1.930
hsa04210 Apoptosis	-1.810
hsa04670 Leukocyte transendothelial migration	-1.778
hsa00564 Glycerophospholipid metabolism	-1.619
hsa04010 MAPK signaling pathway	-1.582
hsa04920 Adipocytokine signaling pathway	-1.503
hsa04810 Regulation of actin cytoskeleton	-1.477
hsa00910 Nitrogen metabolism	-1.415
hsa04144 Endocytosis	-1.330
hsa04140 Regulation of autophagy	-1.328
hsa04720 Long-term potentiation	-1.299
hsa04350 TGF-beta signaling pathway	-1.246
hsa04666 Fc gamma R-mediated phagocytosis	-1.226
hsa04380 Osteoclast differentiation	-1.228
hsa04270 Vascular smooth muscle contraction	-1.199
hsa04070 Phosphatidylinositol signaling system	-1.197
hsa00760 Nicotinate and nicotinamide metabolism	-1.163
hsa04360 Axon guidance	-1.076
hsa04664 Fc epsilon RI signaling pathway	-1.042
hsa00562 Inositol phosphate metabolism	-1.020
hsa04310 Wnt signaling pathway	-1.001
hsa04966 Collecting duct acid secretion	-1.003
hsa00601 Glycosphingolipid biosynthesis - lacto and neolacto series	-0.997
hsa04730 Long-term depression	-0.965
hsa04630 Jak-STAT signaling pathway	-0.938
hsa00770 Pantothenate and CoA biosynthesis	-0.912
hsa04370 VEGF signaling pathway	-0.894
hsa04120 Ubiquitin mediated proteolysis	-0.860
hsa04520 Adherens junction	-0.811
hsa04012 ErbB signaling pathway	-0.809
hsa04115 p53 signaling pathway	-0.805
hsa04662 B cell receptor signaling pathway	-0.768
hsa04976 Bile secretion	-0.723
hsa04062 Chemokine signaling pathway	-0.699
hsa00563 Glycosylphosphatidylinositol(GPI)-anchor biosynthesis	-0.660
hsa00140 Steroid hormone biosynthesis	-0.643
hsa04530 Tight junction	-0.630
hsa04640 Hematopoietic cell lineage	-0.627
hsa00512 Mucin type O-Glycan biosynthesis	-0.596
hsa00590 Arachidonic acid metabolism	-0.570
hsa00430 Taurine and hypotaurine metabolism	-0.557
hsa02010 ABC transporters	-0.544
hsa04510 Focal adhesion	-0.537
hsa04742 Taste transduction	-0.522
hsa04540 Gap junction	-0.518
hsa04971 Gastric acid secretion	-0.515
hsa04910 Insulin signaling pathway	-0.515
hsa00260 Glycine, serine and threonine metabolism	-0.508
hsa04614 Renin-angiotensin system	-0.507
hsa04330 Notch signaling pathway	-0.470
hsa04660 T cell receptor signaling pathway	-0.467
hsa04975 Fat digestion and absorption	-0.437
hsa04912 GnRH signaling pathway	-0.432
hsa04150 mTOR signaling pathway	-0.426
hsa04340 Hedgehog signaling pathway	-0.412
hsa00533 Glycosaminoglycan biosynthesis - keratan sulfate	-0.415
hsa03022 Basal transcription factors	-0.408
hsa04970 Salivary secretion	-0.399
hsa00860 Porphyrin and chlorophyll metabolism	-0.380
hsa04964 Proximal tubule bicarbonate reclamation	-0.377
hsa04710 Circadian rhythm - mammal	-0.342
hsa00790 Folate biosynthesis	-0.346
hsa04620 Toll-like receptor signaling pathway	-0.331
hsa04320 Dorso-ventral axis formation	-0.300
hsa00591 Linoleic acid metabolism	-0.292
hsa04622 RIG-I-like receptor signaling pathway	-0.276

The most striking finding in pathways differentially regulated at the site of disease is the emphasis on multiple metabolic pathways. There is concerted upregulation of pathways important in central carbon metabolism (e.g. glyoxylate and dicarboxylate metabolism, citrate cycle, propanoate metabolism), pathways critical to growth in *M. tuberculosis*. Previous *in vitro* findings suggest alignment of MTB metabolism with its current environment [164] as well as large-scale manipulation of host metabolism by trehalose dimycolate, a mycobacterial cell wall lipid [165]. The latter results in dysregulated host lipid metabolism, a finding associated with caseation of granulomas.

In addition to large-scale metabolic rearrangements, there is upregulation of transcriptional and translational machinery, indicating site-specific protein synthesis, possibly in direct response to the infection. Upregulated DNA duplication pathways suggest cell proliferation at the site of disease. Glycan biosynthesis is also upregulated. As glycans form an important part of mycobacterial cell walls, one can speculate whether this host pathway is actively utilised by MTB to meet its requirements.

Upregulation of the proteasome pathway is probably in part related to upregulated MHC class I activity, as the proteasome plays a critical role in the function of the adaptive immune system by producing peptides to display on MHC class I molecules [166].

Downregulated pathways are also of significant interest. Natural killer cell cytotoxicity is strongly downregulated, perhaps indicating a defensive strategy by the mycobacterium. Other pathways which intuitively should be upregulated in a protective response are leukocyte transendothelial migration, Fc- gamma receptor mediated phagocytosis, regulation of autophagy (both ATG-1 and ATG-8 are downregulated), again hinting at the possibility that the immune response at the site of disease is inappropriately downregulated.

Finally, the overlap of upregulated and downregulated pathways in HIV-1 uninfected and -infected individuals is near complete with additional pathways dysregulated in HIV-1 infected individuals (Figure 12.6). Surprisingly, there are no pathways that are regulated in opposite directions depending on HIV status.

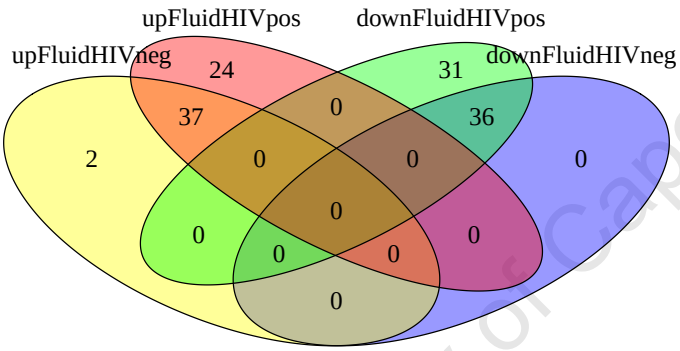
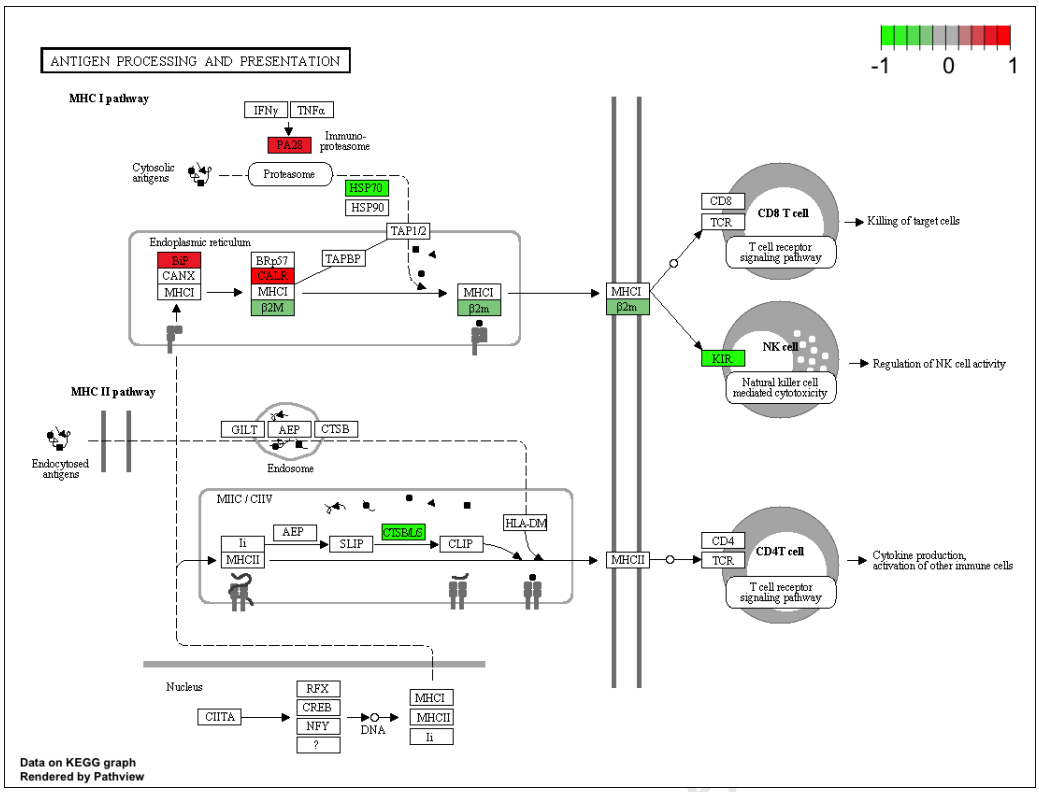


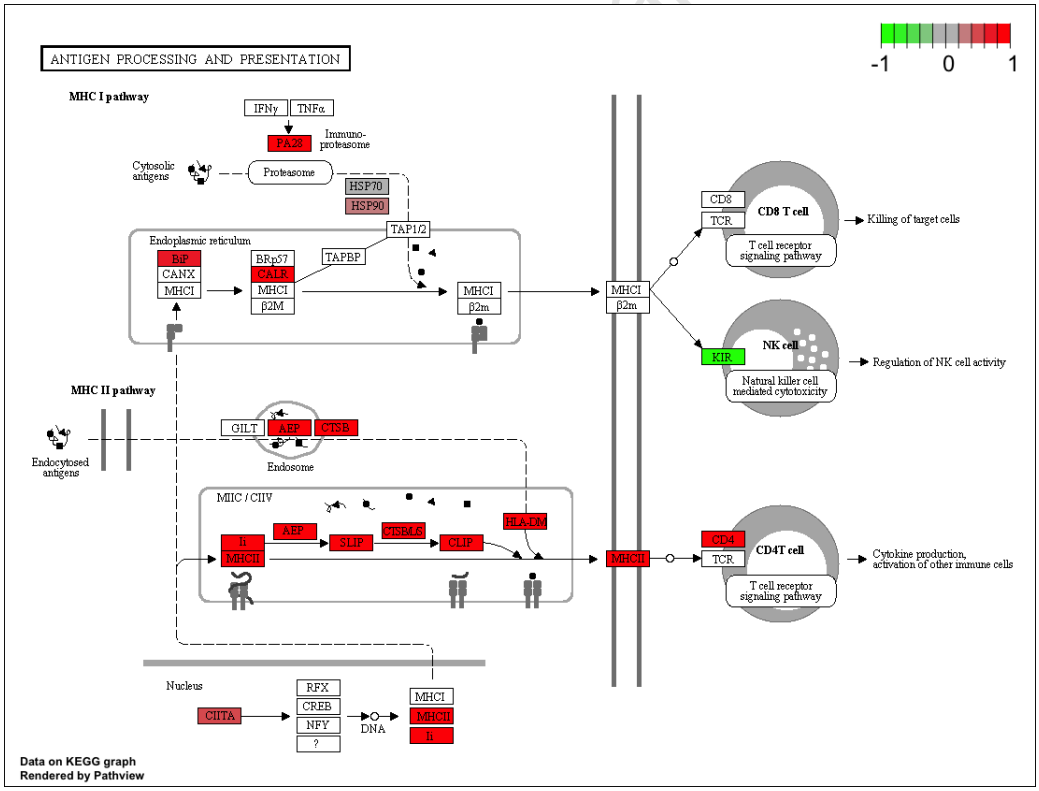
Figure 12.6: **Overlap of differentially regulated pathways stratified by HIV status**

12.2.1 Pathway visualisation

Figure 12.7 shows the KEGG pathway *Antigen processing and presentation*. This pathway is universally upregulated in pericardial fluid in individuals with tuberculous pericarditis, but its effects may be different depending on HIV-1 infection status. Regardless of HIV status the pathway *Natural killer cell-mediated cytotoxicity* is strongly downregulated, raising questions about mechanisms involved in this counterintuitive phenomenon. One question that is suggested by the data is whether downregulation of NK cell cytotoxicty is required for the pathogenesis of tuberculosis.

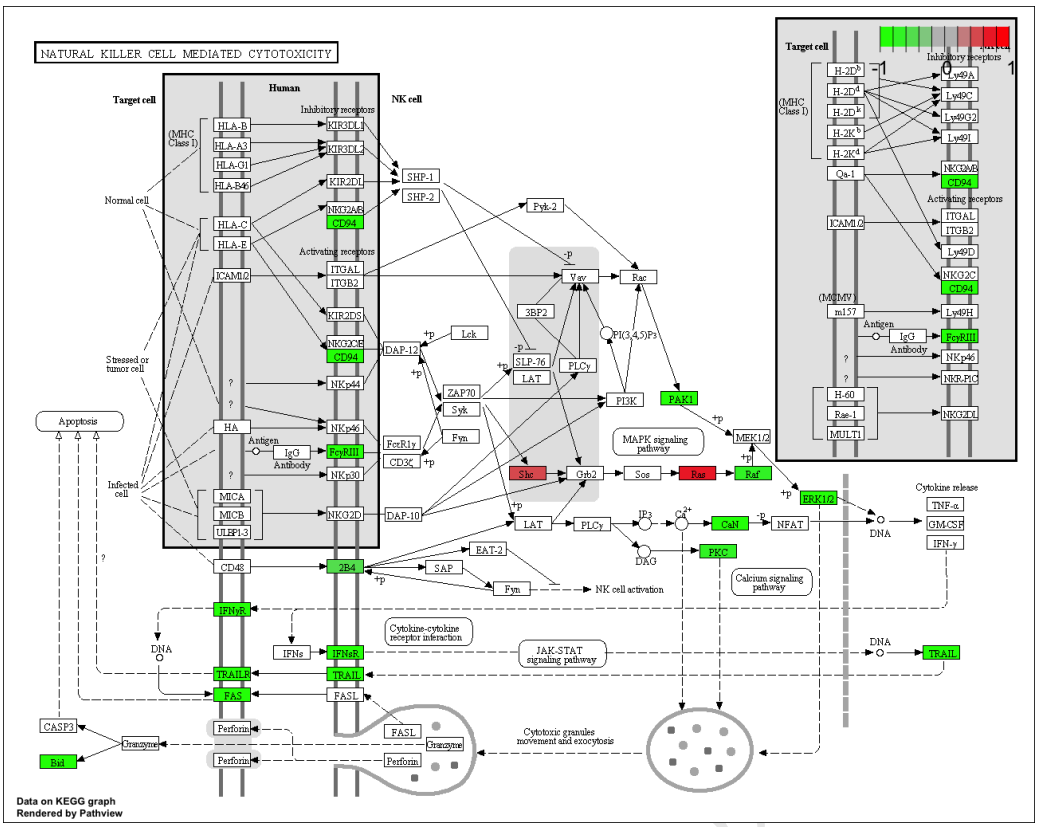


(a) **HIV-1 uninfected**

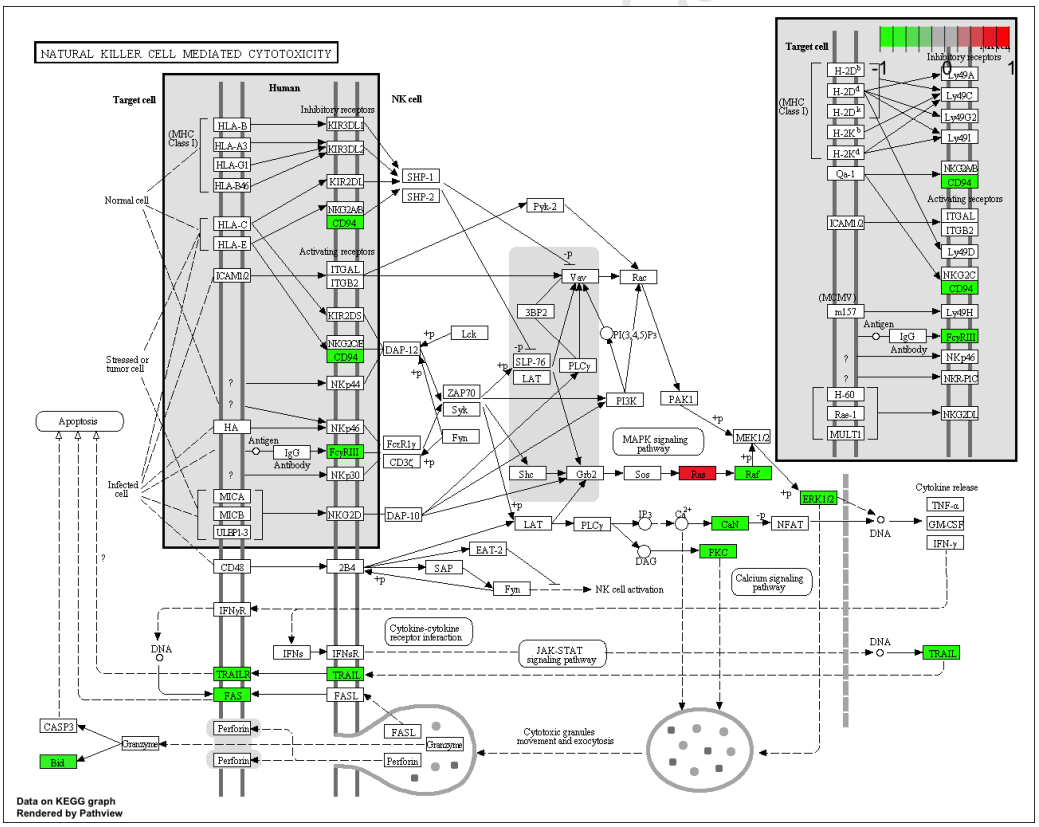


(b) **HIV-1 infected**

Figure 12.7: **Antigen processing and presentation pathway.** Canonical pathway *hsa04612* from KEGG, rendered by *pathview* software. The logfold change estimates output by limma are overlaid, normalised to a range -1 to +1. Colours: *red* = upregulated in pericardial fluid; *green* = downregulated in pericardial fluid.



(a) **HIV-1 uninfected**



(b) **HIV-1 infected**

Figure 12.8: **Natural killer cell mediated cytotoxicity.** Canonical pathway *hsa04650* from KEGG, rendered by *pathview* software. The logfold change estimates output by limma are overlaid, normalised to a range -1 to +1. Colours: *red* = upregulated in pericardial fluid; *green* = downregulated in pericardial fluid.

Part V

Overall conclusions

13 Conclusions and future work

13.1 Summary of novel results

Below I summarise the novel aspects of this work:

1. Creation of a unique and comprehensive microarray dataset (with detailed phenotype information) enabling investigation of multiple aspects of HIV-TB in blood and the site of disease. To my knowledge similar datasets do not exist.

2. Development of a comprehensive analysis pipeline using only open-source software, enabling research reproducibility once the dataset and analysis pipeline are jointly published. In addition to standard differential expression analysis, this pipeline also extracts information regarding cell proportions, modular organisation of co-expressed genes and pathway enrichment, resulting in a comprehensive overview of biological phenomena in the system under study.

3. Application of the analytic pipeline to multiple data subsets which are placed in clear relation to each other using the hypercube analogy that describes sample embedding in phenotype space. This maximises the information that can be extracted from the complex signals embedded in the data.

4. New insights into the biology of HIV-TB were attained using the above methods, in particular the strong suggestion that NK cells may play an important role in tuberculosis and the transcriptional response a the site of disease resulting in major metabolic rearrangements that may support mycobacterial growth.

13.2 An approach to heterogeneous microarray data

Biology is complicated. Acknowledging this complexity in experimental designs opens the door to potentially deep insights, at the price of significantly increased analytic difficulty. Given the potential for interactions between biological processes that define "phenotype classes" like active tuberculosis or HIV-1 infection, it is important to investigate these processes individually and jointly.

With linear increases in phenotypic complexity of study samples there is a concurrent exponential increase in signal complexity, as shown using the hypercube analogy. To make analysis feasible, the data needs to be understood in relation to the phenotype classes in potentially high-dimensional parameter spaces.

It is tempting to draw conclusions regarding disease pathogenesis from lists of over- and under-expressed transcripts. Care must be taken not to over-interpret results, however. Gene expression is a very complex process, and attempts at measuring this in heterogeneous tissues (blood, pericardial fluid) sampled from a heterogeneous study population will result in data in which multiple signals are embedded in a background influenced by stochastic processes. Therefore, while differential gene expression undoubtedly provides us with the ability to classify samples it also provides us with information directly related to the biological process. Retrieving this "biology signal" from the background remains challenging, but I hope that methods that aim to isolate multiple signals from heterogeneous datasets will allow us understand the biology better.

For the study under discussion, transcriptomes of whole blood and pericardial fluid were established. Each tissue type has a characteristic cellular composition which strongly influences the overall transcript abundance. Differences in transcript abundance between conditions of interest are typically found in studies of this nature. Indeed, differences in transcript abundance may be identified even when comparing groups created randomly, and the challenge becomes identifying biologically relevant differences. This strongly argues for use of a combination of analytic methods that aim to understand the meaning of the tissue (blood and pericardial fluid) transcriptomes.

Transcriptomes vary over time. In this study, a single timepoint was used to generate transcriptomic data. In terms of the two disease groups (pulmonary and pericardial tuberculosis), the timepoint was determined by the study subject presenting to medical care, and undergoing dia-

gnostic workup for tuberculosis. This has the consequence that the disease group may contain significant heterogeneity due to individuals presenting in different stages of the disease (e.g. pulmonary TB with a single disease focus vs. pulmonary tuberculosis with concomitant disseminated disease), potentially confounding the results. Therefore, a "snapshot" transcriptome may yield differences between tuberculosis cases that are simply due to different disease stages, to some extent limiting the generalisability of the results.

13.3 Choice of methods

I chose three methods of analysis: differential expression analysis using linear models, deconvolution and cell-type specific differential expression, and weighted gene co-expression network analysis. Results were put into biological context using pathway analysis. The three methods complement each other, as different features of the the biological problem are explored. Together, these methods provide results that may serve as input data for further analyses.

13.4 Results

Expectedly, the largest differences in gene expression can be found when comparing the site of disease to blood. This is mainly due to different proportions of cell types in the two compartments, but cell-type specific differential expression strongly suggests that within multiple cell types biologically relevant differences can be identified. Furthermore, the site of disease exhibits markedly different metabolic activity when compared to blood. It remains to be determined which of these metabolic alterations is caused by *Mycobacterium tuberculosis* to facilitate its survival by creating an environment optimised for its own growth, which are of a defensive nature, and which are unrelated to the presence of the bacterium. Finally, several pathways thought to play a protective role in the response to *M. tuberculosis* are downregulated at the site of disease, possibly indicating additional defence mechanisms of the pathogen.

The effect of HIV-1 infection on transcriptional profiles is a lot more subtle than the effect of compartment. For instance, few or no probes were significantly differentially expressed in individuals with or without concurrent active tuberculosis. This is puzzling, and may be a result of

small numbers of HIV-1 uninfected samples in the various analyses investigating this contrast. Despite this, the top 300 and top 500 differentially regulated probes do successfully cluster the input data, and cell-type specific differential gene expression is detectable. Pathway analysis recapitulates some key results (e.g. differential regulation in the RIG-1-like pathway), and also show that the differential regulation of pathways in HIV-1 infection is strongly influenced by concurrent active tuberculosis.

The transcriptional response to active tuberculosis is to a large extent conserved regardless of HIV-1 infection status. This response consists of upregulation of pathways central to energy metabolism, phagocytosis and downstream events and innate immune responses (like Toll-like receptor signalling). In HIV-1 co-infection, a number of additional pathways are up- or downregulated, indicating that active tuberculosis and HIV-1 interact on many levels.

In each of the analyses, modules were detected using weighted gene co-expression network analysis. These modules have not been characterised on a functional level (see *Future Work*, below), but is is interesting to note that module size varies widely. What do these modules represent? While the answer to this is not yet known, I speculate that study of the modules may reveal pathway groups, expressed in concert, given that some modules contain thousands of genes. In some cases, the modules may represent an even higher level of organisation, namely cell types.

The strategy employed in this analysis has afforded us a view of different aspects contributing to HIV-TB pathobiology. In all three main contrasts, a common dual theme emerges: disease (be it HIV-1 infection or tuberculosis) is associated with striking alterations in immune system function at many levels as well as significant changes in host metabolism. While this has been known for a long time, the results presented here in this thesis give a detailed look at the idea that host metabolism and immune response are part of a higher-order phenomenon: that of pathogens utilising humans as an ecologic niche by evading immune responses and modifying host metabolism to serve their respective needs. This opens up new avenues of investigation, in which immune responses and host metabolic response are assessed as one functional unit. The pathogens' reliance on host metabolic pathways may well turn out to be their Achilles heel, and therefore I argue that host metabolic responses should be studied with the potential of developing new therapeutic strategies.

As a final word on this, as a clinician I would like to state that these results should not come as a big surprise. On a whole system level (i.e. patient physiology) we are deeply familiar with

the metabolic consequences of HIV-1 infection (the entity "wasting disease" is AIDS defining, after all) as well as tuberculosis (or "consumption", a term no longer in widespread use). Maybe the focus of future investigations should shift to include mechanisms of wholesale manipulation of energy metabolism by both pathogens, in addition to defining mechanisms leading to immune system evasion and immunopathology. This addresses the following speculative scenario: what if the contribution that HIV-1 infection makes to the increased risk for developing active TB is not immune dysfunction alone (although this doubtlessly plays a crucial role) but also re-engineering of host metabolism to better suit *M. tuberculosis* growth and survival.

13.5 Future work

13.5.1 Informatics

This thesis presents only a few results for a subset of all contrasts examined. The remaining results, constituting a dataset in their own right require ongoing analysis and interpretation. Specifically, the modules identified may point to ways how various pathways are connected and co-expressed. I have shown that some of the results, obtained in a deep sweep of transcriptional activity in active TB and HIV-1 infection replicate previously published findings. I intend to match all results of this analysis with the extant TB literature. The potential outcome for each result is threefold:

1. Agreement with published results serving as validation of prior knowledge
2. Disagreement with published results serving as potential alternative hypothesis which may require experimental assessment
3. Novel finding, indicating the need for validation experiments

I plan to prepare a catalogue of targeted experiments based on the above analysis which may aid our understanding of HIV-1 infection, tuberculosis and co-infection.

In addition to the present dataset, the opportunity to re-analyse existing datasets using the pipeline presented here will be utilised, as several groups have published data that is freely accessible.

Ongoing efforts will be made to optimise the analytic pipeline. Pathway analysis, currently implemented in separate code, will be integrated into the pipeline, and the entire pipeline will be upgraded to function under the latest release of R and bioconductor.

13.5.2 Validation

All results of this analysis are at the RNA level. Validation experiments are required to assess these results in the context of protein-level and cellular level findings. To start this process, targeted experiments will be designed to utilise stored serum, cell-free pericardial fluid and frozen, unstimulated PBMC and pericardial fluid cells. Validation will include, but not be limited to assessment of cell proportions by flow cytometry, and comparing these to the results obtained by deconvolution, as well as determination of key proteins (cytokines, chemokines, enzymes) in the various contexts. An additional class of validation experiments may be performed at the metabolite level, in order to define the metabolic niche utilised by *M. tuberculosis*.

13.5.3 New projects

Not all potential contrasts could be examined; Figure 8.1 illustrates this graphically. This framework can be used to design a new study comparing blood and site of disease samples even in LTBI, PTB and possibly other forms of tuberculosis, in contexts with or without HIV-1 co-infection. This may be achieved by careful study design and stratified sampling to ensure that all vertices in high-dimensional phenotype space are as evenly populated as possible.

Bibliography

1. Deffur, A., Mulder, N. J. & Wilkinson, R. J. Co-infection with Mycobacterium tuberculosis and human immunodeficiency virus: an overview and motivation for systems approaches. *Pathogens and Disease* (July 2013).

2. Tiemersma, E. W., van der Werf, M. J., Borgdorff, M. W., Williams, B. G. & Nagelkerke, N. J. D. Natural history of tuberculosis: duration and fatality of untreated pulmonary tuberculosis in HIV negative patients: a systematic review. *PLoS ONE* **6,** e17601 (2011).

3. Kwan, C. K. & Ernst, J. D. HIV and Tuberculosis: a Deadly Human Syndemic. *Clinical Microbiology Reviews* **24,** 351–376 (Apr. 2011).

4. Corbett, E. L. *et al.* Epidemiology of tuberculosis in a high HIV prevalence population provided with enhanced diagnosis of symptomatic disease. *PLoS Medicine* **4,** e22 (Jan. 2007).

5. Sonnenberg, P. *et al.* How soon after infection with HIV does the risk of tuberculosis start to increase? A retrospective cohort study in South African gold miners. *J Infect Dis* **191,** 150–158 (Jan. 2005).

6. Frieden, T. R., Sterling, T. R., Munsiff, S. S., Watt, C. J. & Dye, C. Tuberculosis. *Lancet* **362,** 887–899 (Sept. 2003).

7. Daley, C. L. *et al.* An outbreak of tuberculosis with accelerated progression among persons infected with the human immunodeficiency virus. An analysis using restriction-fragment-length polymorphisms. *New England Journal of Medicine* **326,** 231–235 (Jan. 1992).

8. Selwyn, P. A. *et al.* A prospective study of the risk of tuberculosis among intravenous drug users with human immunodeficiency virus infection. *New England Journal of Medicine* **320,** 545–550 (Mar. 1989).

9. Sonnenberg, P. *et al.* HIV-1 and recurrence, relapse, and reinfection of tuberculosis after cure: a cohort study in South African mineworkers. *Lancet* **358,** 1687–1693 (Nov. 2001).

10. Charalambous, S. *et al.* Contribution of reinfection to recurrent tuberculosis in South African gold miners. *The International Journal of Tuberculosis and Lung Disease* **12,** 942–948 (Aug. 2008).

11. Sterling, T. R., Pham, P. A. & Chaisson, R. E. HIV infection-related tuberculosis: clinical manifestations and treatment. *Clinical Infectious Diseases* **50 Suppl 3,** S223–30 (May 2010).

12. Schutz, C., Meintjes, G., Almajid, F., Wilkinson, R. J. & Pozniak, A. Clinical management of tuberculosis and HIV-1 co-infection. *The European Respiratory Journal* **36,** 1460–1481 (Dec. 2010).

13. Oni, T. *et al.* High prevalence of subclinical tuberculosis in HIV-1-infected persons without advanced immunodeficiency: implications for TB screening. *Thorax* **66,** 669–673 (Aug. 2011).

14. Rangaka, M. X. *et al.* Interferon release does not add discriminatory value to smear-negative HIV-tuberculosis algorithms. *The European Respiratory Journal* **39,** 163–171 (Jan. 2012).

15. Jones, B. E. *et al.* Relationship of the manifestations of tuberculosis to CD4 cell counts in patients with human immunodeficiency virus infection. *The American Review of Respiratory Disease* **148,** 1292–1297 (Nov. 1993).

16. Gonzalez, O. Y. *et al.* Extra-pulmonary manifestations in a large metropolitan area with a low incidence of tuberculosis. *The International Journal of Tuberculosis and Lung Disease* **7,** 1178–1185 (Dec. 2003).

17. DeRiemer, K., Kawamura, L. M., Hopewell, P. C. & Daley, C. L. Quantitative impact of human immunodeficiency virus infection on tuberculosis dynamics. *American Journal of Respiratory and Critical Care Medicine* **176,** 936–944 (Nov. 2007).

18. Andrews, J. R., Wood, R., Bekker, L.-G., Middelkoop, K. & Walensky, R. P. Projecting the benefits of antiretroviral therapy for HIV prevention: the impact of population mobility and linkage to care. *J Infect Dis* **206,** 543–551 (Aug. 2012).

19. Barry, C. E. *et al.* The spectrum of latent tuberculosis: rethinking the biology and intervention strategies. *Nature Reviews Microbiology* **7,** 845–855 (Dec. 2009).

20. Goo, J. M. *et al.* Pulmonary tuberculoma evaluated by means of FDG PET: findings in 10 cases. *Radiology* **216,** 117–121 (July 2000).

21. Hara, T., Kosaka, N., Suzuki, T., Kudo, K. & Niino, H. Uptake Rates of 18F-Fluorodeoxyglucose and 11C-Choline in Lung Cancer and Pulmonary Tuberculosis A Positron Emission Tomography Study. *Chest* **124,** 893–901 (2003).

22. Chiang, C.-Y. & Riley, L. W. Exogenous reinfection in tuberculosis. *The Lancet Infectious Diseases* **5,** 629–636 (Oct. 2005).

23. Andrews, J. R. *et al.* Risk of progression to active tuberculosis following reinfection with *Mycobacterium tuberculosis*. *Clinical Infectious Diseases* **54,** 784–791 (Mar. 2012).

24. Lawn, S. D., Myer, L., Edwards, D., Bekker, L.-G. & Wood, R. Short-term and long-term risk of tuberculosis associated with CD4 cell recovery during antiretroviral therapy in South Africa. *AIDS* **23,** 1717 (2009).

25. Meintjes, G. *et al.* Corticosteroid-modulated immune activation in the tuberculosis immune reconstitution inflammatory syndrome. *American Journal of Respiratory and Critical Care Medicine* **186,** 369–377 (Aug. 2012).

26. Bourgarit, A. *et al.* Explosion of tuberculin-specific Th1-responses induces immune restoration syndrome in tuberculosis and HIV co-infected patients. *AIDS* **20,** F1–7 (Jan. 2006).

27. Meintjes, G. *et al.* Type 1 helper T cells and FoxP3-positive T cells in HIV-tuberculosis-associated immune reconstitution inflammatory syndrome. *American Journal of Respiratory and Critical Care Medicine* **178,** 1083–1089 (Nov. 2008).

28. Seddiki, N. *et al.* Proliferation of weakly suppressive regulatory CD4+ T cells is associated with over-active CD4+ T-cell responses in HIV-positive patients with mycobacterial immune restoration disease. *European Journal of Immunology* **39,** 391–403 (Feb. 2009).

29. Lawn, S. D., Wainwright, H. & Orrell, C. Fatal unmasking tuberculosis immune reconstitution disease with bronchiolitis obliterans organizing pneumonia: the role of macrophages. *AIDS* **23,** 143–145 (Jan. 2009).

30. Tadokera, R. *et al.* Hypercytokinaemia accompanies HIV-tuberculosis immune reconstitution inflammatory syndrome. *The European Respiratory Journal* **37,** 1248–1259 (May 2011).

31. Pean, P. *et al.* Natural killer cell degranulation capacity predicts early onset of the immune reconstitution inflammatory syndrome (IRIS) in HIV-infected patients with tuberculosis. *Blood* **119,** 3315–3320 (Apr. 2012).

32. Bourgarit, A. *et al.* Tuberculosis-associated immune restoration syndrome in HIV-1-infected patients involves tuberculin-specific CD4 Th1 cells and KIR-negative gammadelta T cells. *Journal of Immunology* **183,** 3915–3923 (Sept. 2009).

33. Wilkinson, K. A. *et al.* Dissection of regenerating T-Cell responses against tuberculosis in HIV-infected adults sensitized by *Mycobacterium tuberculosis*. *American Journal of Respiratory and Critical Care Medicine* **180,** 674–683 (Oct. 2009).

34. Robbins, G. K. *et al.* Incomplete reconstitution of T cell subsets on combination antiretroviral therapy in the AIDS Clinical Trials Group protocol 384. *Clinical Infectious Diseases* **48,** 350–361 (Feb. 2009).

35. O'Garra, A. *et al.* The immune response in tuberculosis. *Annual Review of Immunology* **31,** 475–527 (Mar. 2013).

36. Haase, A. T. Population biology of HIV-1 infection: viral and CD4+ T cell demographics and dynamics in lymphatic tissues. *Annual Review of Immunology* **17,** 625–656 (1999).

37. Mehandru, S. *et al.* Primary HIV-1 infection is associated with preferential depletion of CD4+ T lymphocytes from effector sites in the gastrointestinal tract. *The Journal of Experimental Medicine* **200,** 761–770 (Sept. 2004).

38. Clerici, M. *et al.* Detection of three distinct patterns of T helper cell dysfunction in asymptomatic, human immunodeficiency virus-seropositive patients. Independence of CD4+ cell numbers and clinical staging. *The Journal of Clinical Investigation* **84,** 1892–1899 (Dec. 1989).

39. Sutherland, R. *et al.* Impaired IFN-gamma-secreting capacity in mycobacterial antigen-specific CD4 T cells during chronic HIV-1 infection despite long-term HAART. *AIDS* **20,** 821–829 (Apr. 2006).

40. Kalsdorf, B. *et al.* HIV-1 infection impairs the bronchoalveolar T-cell response to mycobacteria. *American Journal of Respiratory and Critical Care Medicine* **180,** 1262–1270 (Dec. 2009).

41. Hammond, A. S. *et al.* Mycobacterial T cell responses in HIV-infected patients with advanced immunosuppression. *J Infect Dis* **197,** 295–299 (Jan. 2008).

42. Geldmacher, C. *et al.* Early depletion of *Mycobacterium tuberculosis*-specific T helper 1 cell responses after HIV-1 infection. *J Infect Dis* **198,** 1590–1598 (Dec. 2008).

43. Cho, S. *et al.* Antimicrobial activity of MHC class I-restricted CD8+ T cells in human tuberculosis. *Proceedings of the National Academy of Sciences of the United States of America* **97,** 12210–12215 (Oct. 2000).

44. Mogues, T., Goodrich, M. E., Ryan, L., LaCourse, R. & North, R. J. The relative importance of T cell subsets in immunity and immunopathology of airborne *Mycobacterium tuberculosis* infection in mice. *The Journal of Experimental Medicine* **193,** 271–280 (Feb. 2001).

45. Chen, C. Y. *et al.* A critical role for CD8 T cells in a nonhuman primate model of tuberculosis. *PLoS Pathogens* **5,** e1000392 (Apr. 2009).

46. Bruns, H. *et al.* Anti-TNF immunotherapy reduces CD8+ T cell-mediated antimicrobial activity against *Mycobacterium tuberculosis* in humans. *The Journal of Clinical Investigation* **119,** 1167–1177 (May 2009).

47. Toossi, Z. *et al.* Impact of tuberculosis (TB) on HIV-1 activity in dually infected patients. *Clinical and Experimental Immunology* **123,** 233–238 (Feb. 2001).

48. Collins, K. R., Quiñones-Mateu, M. E., Toossi, Z. & Arts, E. J. Impact of tuberculosis on HIV-1 replication, diversity, and disease progression. *AIDS reviews* **4,** 165–176 (July 2002).

49. Matthews, K. *et al.* HIV-1 infection alters CD4+ memory T-cell phenotype at the site of disease in extrapulmonary tuberculosis. *European Journal of Immunology* **42,** 147–157 (Jan. 2012).

50. Hoshino, Y. *et al. Mycobacterium tuberculosis*-induced CXCR4 and chemokine expression leads to preferential X4 HIV-1 replication in human macrophages. *Journal of Immunology* **172,** 6251–6258 (May 2004).

51. Toossi, Z. *et al.* Increased replication of HIV-1 at sites of *Mycobacterium tuberculosis* infection: potential mechanisms of viral activation. *Journal of Acquired Immune Deficiency Syndromes* **28,** 1–8 (Sept. 2001).

52. Hoshino, Y. *et al.* Mechanisms of polymorphonuclear neutrophil-mediated induction of HIV-1 replication in macrophages during pulmonary tuberculosis. *J Infect Dis* **195,** 1303–1310 (May 2007).

53. Toossi, Z. *et al.* Activation of P-TEFb at sites of dual HIV/TB infection, and inhibition of MTB-induced HIV transcriptional activation by the inhibitor of CDK9, Indirubin-3'-monoxime. *AIDS Research and Human Retroviruses* **28,** 182–187 (Feb. 2012).

54. Badri, M., Ehrlich, R., Wood, R., Pulerwitz, T. & Maartens, G. Association between tuberculosis and HIV disease progression in a high tuberculosis prevalence area. *The International Journal of Tuberculosis and Lung Disease* **5,** 225–232 (Mar. 2001).

55. Mañas, E. *et al.* Impact of tuberculosis on the course of HIV-infected patients with a high initial CD4 lymphocyte count. *The International Journal of Tuberculosis and Lung Disease* **8,** 451–457 (Apr. 2004).

56. Moller, M., de Wit, E. & Hoal, E. G. Past, present and future directions in human genetic susceptibility to tuberculosis. *FEMS Immunology and Medical Microbiology* **58,** 3–26 (Feb. 2010).

57. Barreiro, L. B. *et al.* Deciphering the genetic architecture of variation in the immune response to *Mycobacterium tuberculosis* infection. *Proceedings of the National Academy of Sciences of the United States of America* **109,** 1204–1209 (Jan. 2012).

58. 1000 Genomes Project Consortium *et al.* An integrated map of genetic variation from 1,092 human genomes. *Nature* **491,** 56–65 (Nov. 2012).

59. Thye, T. *et al.* Common variants at 11p13 are associated with susceptibility to tuberculosis. *Nature genetics* **44,** 257–259 (Mar. 2012).

60. Pontillo, A. *et al.* Susceptibility to *Mycobacterium tuberculosis* Infection in HIV-Positive Patients Is Associated With CARD8 Genetic Variant. *Journal of Acquired Immune Deficiency Syndromes* **63,** 147–151 (June 2013).

61. Ramaseri Sunder, S. *et al.* IL-10 high producing genotype predisposes HIV infected individuals to TB infection. *Hum Immunol* **73,** 605–611 (June 2012).

62. Stein, C. M. Genetic epidemiology of tuberculosis susceptibility: impact of study design. *PLoS Pathogens* **7,** e1001189 (2011).

63. Flynn, J. L. Lessons from experimental *Mycobacterium tuberculosis* infections. *Microbes Infect* **8,** 1179–1188 (Apr. 2006).

64. Berges, B. K. & Rowan, M. R. The utility of the new generation of humanized mice to study HIV-1 infection: transmission, prevention, pathogenesis, and treatment. *Retrovirology* **8,** 65 (2011).

65. Diedrich, C. R. & Flynn, J. L. HIV-1/*Mycobacterium tuberculosis* coinfection immunology: how does HIV-1 exacerbate tuberculosis? *Infection and Immunity* **79,** 1407–1417 (Apr. 2011).

66. Hanna, Z. *et al.* Nef harbors a major determinant of pathogenicity for an AIDS-like disease induced by HIV-1 in transgenic mice. *Cell* **95,** 163–175 (Oct. 1998).

67. Hanna, Z. *et al.* Transgenic mice expressing human immunodeficiency virus type 1 in immune cells develop a severe AIDS-like disease. *Journal of Virology* **72,** 121–132 (Jan. 1998).

68. Hanna, Z. *et al.* Selective expression of human immunodeficiency virus Nef in specific immune cell populations of transgenic mice is associated with distinct AIDS-like phenotypes. *Journal of Virology* **83,** 9743–9758 (Oct. 2009).

69. Denton, P. W. *et al.* Antiretroviral pre-exposure prophylaxis prevents vaginal transmission of HIV-1 in humanized BLT mice. *PLoS Medicine* **5,** e16 (Jan. 2008).

70. Denton, P. W. *et al.* Systemic administration of antiretrovirals prior to exposure prevents rectal and intravenous HIV-1 transmission in humanized BLT mice. *PLoS ONE* **5,** e8829 (2010).

71. Sun, Z. *et al.* Intrarectal transmission, systemic infection, and CD4+ T cell depletion in humanized mice infected with HIV-1. *The Journal of Experimental Medicine* **204,** 705–714 (Apr. 2007).

72. Orme, I. M. The mouse as a useful model of tuberculosis. *Tuberculosis* **83,** 112–115 (2003).

73. Seok, J. *et al.* Genomic responses in mouse models poorly mimic human inflammatory diseases. *Proceedings of the National Academy of Sciences of the United States of America* **110,** 3507–3512 (Feb. 2013).

74. Flynn, J. L. *et al.* Non-human primates: a model for tuberculosis research. *Tuberculosis* **83,** 116–118 (2003).

75. Kaushal, D., Mehra, S., Didier, P. J. & Lackner, A. A. The non-human primate model of tuberculosis. *Journal of Medical Primatology* **41,** 191–201 (June 2012).

76. Gagneux, S. Host-pathogen coevolution in human tuberculosis. *Philosophical Transactions of the Royal Society of London Series B, Biological Sciences* **367,** 850–859 (Mar. 2012).

77. Gagneux, S. & Small, P. M. Global phylogeography of *Mycobacterium tuberculosis* and implications for tuberculosis product development. *The Lancet Infectious Diseases* **7,** 328–337 (May 2007).

78. Ryan, F. P. Human endogenous retroviruses in health and disease: a symbiotic perspective. *Journal of the Royal Society of Medicine* **97,** 560–565 (Dec. 2004).

79. Gillespie, S. H. Evolution of drug resistance in *Mycobacterium tuberculosis*: clinical and molecular perspective. *Antimicrob Agents Chemother* **46,** 267–274 (Feb. 2002).

80. Fenner, L. *et al.* HIV Infection Disrupts the Sympatric Host-Pathogen Relationship in Human Tuberculosis. *PLoS Genetics* **9,** e1003318 (Mar. 2013).

81. Zak, D. E. & Aderem, A. Systems biology of innate immunity. *Immunological Reviews* **227,** 264–282 (Jan. 2009).

82. Berry, M. P. R. *et al.* An interferon-inducible neutrophil-driven blood transcriptional signature in human tuberculosis. *Nature* **466,** 973–977 (Aug. 2010).

83. Chaussabel, D. *et al.* A Modular Analysis Framework for Blood Genomics Studies: Application to Systemic Lupus Erythematosus. *Immunity* **29,** 150–164 (July 2008).

84. Chaussabel, D., Pascual, V. & Banchereau, J. Assessing the human immune system through blood transcriptomics. *BMC Biology* **8,** 84 (2010).

85. Maertzdorf, J. *et al.* Human gene expression profiles of susceptibility and resistance in tuberculosis. *Genes and Immunity* **12,** 15–22 (Jan. 2011).

86. Ottenhoff, T. H. M. *et al.* Genome-wide expression profiling identifies type 1 interferon response pathways in active tuberculosis. *PLoS ONE* **7,** e45839 (2012).

87. Bloom, C. I. *et al.* Detectable changes in the blood transcriptome are present after two weeks of antituberculosis therapy. *PLoS ONE* **7,** e46191 (2012).

88. Koh, G. C. K. W. *et al.* Host responses to melioidosis and tuberculosis are both dominated by interferon-mediated signaling. *PLoS ONE* **8,** e54961 (2013).

89. Mazandu, G. K. & Mulder, N. J. Scoring protein relationships in functional interaction networks predicted from sequence data. *PLoS ONE* **6,** e18607 (2011).

90. Mazandu, G. K., Opap, K. & Mulder, N. J. Contribution of microarray data to the advancement of knowledge on the *Mycobacterium tuberculosis* interactome: use of the random partial least squares approach. *Infection, Genetics and Evolution* **11,** 725–733 (June 2011).

91. Magombedze, G. & Mulder, N. Understanding TB latency using computational and dynamic modelling procedures. *Infection, Genetics and Evolution* **13,** 267–283 (Jan. 2013).

92. Young, D., Stark, J. & Kirschner, D. Systems biology of persistent infection: tuberculosis as a case study. *Nature Reviews Microbiology* **6,** 520–528 (July 2008).

93. Marino, S., Linderman, J. J. & Kirschner, D. E. A multifaceted approach to modeling the immune response in tuberculosis. *Wiley interdisciplinary reviews. Systems Biology and Medicine* **3,** 479–489 (July 2011).

94. Marino, S. & Kirschner, D. E. The human immune response to *Mycobacterium tuberculosis* in lung and lymph node. *Journal of Theoretical Biology* **227,** 463–486 (Apr. 2004).

95. Marino, S., El-Kebir, M. & Kirschner, D. A hybrid multi-compartment model of granuloma formation and T cell priming in Tuberculosis. *Journal of Theoretical Biology* **280,** 50–62 (Apr. 2011).

96. Kumar, D. *et al.* Genome-wide analysis of the host intracellular network that regulates survival of *Mycobacterium tuberculosis*. *Cell* **140,** 731–743 (Mar. 2010).

97. Doerks, T., van Noort, V., Minguez, P. & Bork, P. Annotation of the *M. tuberculosis* hypothetical orfeome: adding functional information to more than half of the uncharacterized proteins. *PLoS ONE* **7,** e34302 (2012).

98. Comas, I. & Gagneux, S. A role for systems epidemiology in tuberculosis research. *Trends in Microbiology* **19,** 492–500 (Oct. 2011).

99. Westermann, A. J., Gorski, S. A. & Vogel, J. Dual RNA-seq of pathogen and host. *Nature Reviews Microbiology* **10,** 618–630 (Sept. 2012).

100. Abbas, A. R., Wolslegel, K., Seshasayee, D., Modrusan, Z. & Clark, H. F. Deconvolution of blood microarray data identifies cellular activation patterns in systemic lupus erythematosus. *PLoS ONE* **4,** e6098 (2009).

101. Aderem, A. *et al.* A systems biology approach to infectious disease research: innovating the pathogen-host research paradigm. *mBio* **2,** e00325–10 (2011).

102. World Health Organization. *Global Tuberculosis Report 2012. WHO, Geneva, Switzerland* tech. rep. (Nov. 2012).

103. Russell, J. B. *et al.* Tuberculous effusive-constrictive pericarditis. *Cardiovascular Journal of Africa* **19,** 200–201 (June 2008).

104. Mayosi, B. M. *et al.* Mortality in patients treated for tuberculous pericarditis in sub-Saharan Africa. *South African Medical Journal = Suid-Afrikaanse tydskrif vir geneeskunde* **98,** 36–40 (Jan. 2008).

105. Ntsekhe, M., Wiysonge, C., Volmink, J. A., Commerford, P. J. & Mayosi, B. M. Adjuvant corticosteroids for tuberculous pericarditis: promising, but not proven. *QJM* **96,** 593–599 (Aug. 2003).

106. Mayosi, B. M. *et al.* Clinical characteristics and initial management of patients with tuberculous pericarditis in the HIV era: the Investigation of the Management of Pericarditis in Africa (IMPI Africa) registry. *BMC Infectious Diseases* **6,** 2 (2006).

107. Matthews, K. *Immunological Analysis of Pericardial Tuberculosis* PhD thesis (University of Cape Town, Cape Town, July 2011).

108. R-Core-Team. *R: A Language and Environment for Statistical Computing* R Foundation for Statistical Computing (Vienna, Austria, 2013). <http://www.R-project.org/>.

109. Smyth, G. K. Linear Models and Empirical Bayes Methods for Assessing Differential Expression in Microarray Experiments. *Statistical Applications in Genetics and Molecular Biology* **3** (Jan. 2004).

110. *OOMPA:Overview - MD Anderson Bioinformatics* <http://bioinformatics.mdanderson.org/main/OOMPA:Overview>.

111. Reuter, H., Burgess, L., van Vuuren, W. & Doubell, A. Diagnosing tuberculous pericarditis. *QJM* **99,** 827–839 (Dec. 2006).

112. Matthews, K. *et al.* Predominance of interleukin-22 over interleukin-17 at the site of disease in human tuberculosis. *Tuberculosis* **91,** 587–593 (Nov. 2011).

113. Wolfram Research, I. *Mathematica Edition: Version 9.0* Champaign, Illinois, 2012. <http://www.wolfram.com/>.

114. Illumina, Inc. *Whole-Genome Gene ExpressionDirect HybridizationAssay Guide* (Jan. 2013).

115. Field, L. A. *et al.* Functional identity of genes detectable in expression profiling assays following globin mRNA reduction of peripheral blood samples. *Clinical Biochemistry* **40,** 499–502 (Apr. 2007).

116. Vartanian, K. *et al.* Gene expression profiling of whole blood: comparison of target preparation methods for accurate and reproducible microarray analysis. *BMC Genomics* **10,** 2 (2009).

117. Yang, H. *et al.* Randomization in laboratory procedure is key to obtaining reproducible microarray results. *PLoS ONE* **3,** e3724 (2008).

118. Kuhn, K. *et al.* A novel, high-performance random array platform for quantitative gene expression profiling. *Genome Research* **14,** 2347–2356 (Nov. 2004).

119. Gunderson, K. L. *et al.* Decoding randomly ordered DNA arrays. *Genome Research* **14,** 870–877 (May 2004).

120. Simon, R. M. *et al. Design and Analysis of DNA Microarray Investigations* (Springer, July 2012).

121. Kononen, J. *et al.* Tissue microarrays for high-throughput molecular profiling of tumor specimens. *Nat Med* **4,** 844–847 (July 1998).

122. Sotiriou, C. & Pusztai, L. Gene-Expression Signatures in Breast Cancer. *The New England Journal of Medicine* **360,** 790–800 (Feb. 2009).

123. Schmid, R. *et al.* Comparison of normalization methods for Illumina BeadChip HumanHT-12 v3. *BMC Genomics* **11,** 349 (2010).

124. Du, P., Kibbe, W. A. & Lin, S. M. lumi: a pipeline for processing Illumina microarray. *Bioinformatics* **24,** 1547–1548 (July 2008).

125. Drăghici, S. *Statistics and Data Analysis for Microarrays Using R and Bioconductor* (2012).

126. Benjamini, Y. & Hochberg, Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)* **57,** 289–300 (1995).

127. Reiner, A., Yekutieli, D. & Benjamini, Y. Identifying differentially expressed genes using false discovery rate controlling procedures. *Bioinformatics* **19,** 368–375 (2003).

128. Whitney, A. R. *et al.* Individuality and variation in gene expression patterns in human blood. *Proceedings of the National Academy of Sciences of the United States of America* **100,** 1896–1901 (Feb. 2003).

129. Cobb, J. P. Application of genome-wide expression analysis to human health and disease. *Proceedings of the National Academy of Sciences of the United States of America* **102,** 4801–4806 (Mar. 2005).

130. Palmer, C., Diehn, M., Alizadeh, A. A. & Brown, P. O. Cell-type specific gene expression profiles of leukocytes in human peripheral blood. *BMC Genomics* **7,** 115 (2006).

131. Debey, S. *et al.* Comparison of different isolation techniques prior gene expression profiling of blood derived cells: impact on physiological responses, on overall expression and the role of different cell types. *The Pharmacogenomics Journal* **4,** 193–207 (2004).

132. Feezor, R. J. *et al.* Whole blood and leukocyte RNA isolation for gene expression analyses. *Physiological genomics* **19,** 247–254 (Nov. 2004).

133. Lu, P., Nakorchevskiy, A. & Marcotte, E. M. Expression deconvolution: a reinterpretation of DNA microarray data reveals dynamic changes in cell populations. *Proceedings of the National Academy of Sciences of the United States of America* **100,** 10370 (2003).

134. Gaujoux, R. & Seoighe, C. CellMix: a comprehensive toolbox for gene expression deconvolution. *Bioinformatics* (July 2013).

135. Shen-Orr, S. S. *et al.* Cell type-specific gene expression differences in complex tissues. *Nature Methods* **7,** 287–289 (Apr. 2010).

136. Repsilber, D. *et al.* Biomarker discovery in heterogeneous tissue samples -taking the in-silico deconfounding approach. *BMC Bioinformatics* **11,** 27 (2010).

137. Gong, T. *et al.* Optimal deconvolution of transcriptional profiling data using quadratic programming with application to complex clinical blood samples. *PLoS ONE* **6,** e27156 (2011).

138. Zhong, Y., Wan, Y.-W., Pang, K., Chow, L. M. & Liu, Z. Digital sorting of complex tissues for cell type-specific gene expression profiles. *BMC Bioinformatics* **14,** 89 (2013).

139. Reuter, H., Burgess, L. J., Carstens, M. E. & Doubell, A. F. Characterization of the immunological features of tuberculous pericardial effusions in HIV positive and HIV negative patients in contrast with non-tuberculous effusions. *Tuberculosis* **86,** 125–133 (Mar. 2006).

140. Zhang, B. & Horvath, S. A general framework for weighted gene co-expression network analysis. *Statistical Applications in Genetics and Molecular Biology* **4,** Article17 (2005).

141. Langfelder, P. & Horvath, S. WGCNA: an R package for weighted correlation network analysis. *BMC Bioinformatics* **9,** 559 (2008).

142. Barabasi, A. & Albert, R. Emergence of scaling in random networks. *Science* **286,** 509–512 (Oct. 1999).

143. Albert, R., Jeong, H. & Barabasi, A. Error and attack tolerance of complex networks. *Nature* **406,** 378–382 (July 2000).

144. Jeong, H., Tombor, B., Albert, R., Oltvai, Z. N. & Barabási, A. The large-scale organization of metabolic networks. *Nature* **407,** 651–654 (Oct. 2000).

145. Yook, S.-H., Oltvai, Z. N. & Barabási, A.-L. Functional and topological characterization of protein interaction networks. *Proteomics* **4,** 928–942 (Apr. 2004).

146. Yu, H. *et al.* High-quality binary protein interaction map of the yeast interactome network. *Science* **322,** 104–110 (Oct. 2008).

147. Barabási, A.-L. Scale-free networks: a decade and beyond. *Science* **325,** 412–413 (July 2009).

148. *Extended Overview of Weighted Gene Co-Expression Network Analysis (WGCNA)* (Feb. 2012). <http://labs.genetics.ucla.edu/horvath/CoexpressionNetwork/WORKSHOP/>.

149. Luo, W., Friedman, M. S., Shedden, K., Hankenson, K. D. & Woolf, P. J. GAGE: generally applicable gene set enrichment for pathway analysis. *BMC Bioinformatics* **10,** 161 (2009).

150. Luo, W. & Brouwer, C. Pathview: an R/Bioconductor package for pathway-based data integration and visualization. *Bioinformatics* **29,** 1830–1831 (July 2013).

151. Kanehisa, M. & Goto, S. KEGG: kyoto encyclopedia of genes and genomes. *Nucleic Acids Research* **28,** 27–30 (Jan. 2000).

152. Kanehisa, M., Goto, S., Sato, Y., Furumichi, M. & Tanabe, M. KEGG for integration and interpretation of large-scale molecular data sets. *Nucleic Acids Research* **40,** D109–14 (Jan. 2012).

153. Khatri, P., Sirota, M. & Butte, A. J. Ten years of pathway analysis: current approaches and outstanding challenges. *PLoS Computational Biology* **8,** e1002375 (2012).

154. Hamming, R. W. Error detecting and error correcting codes. *Bell System technical journal* **29,** 147–160 (1950).

155. Menon, R. *et al.* Gender-based blood transcriptomes and interactomes in multiple sclerosis: involvement of SP1 dependent gene transcription. *Journal of autoimmunity* **38,** J144–55 (May 2012).

156. Junqueira-Kipnis, A. P. & Kipnis, A. NK Cells Respond to Pulmonary Infection with Mycobacterium tuberculosis, but Play a Minimal Role in Protection. *The Journal of Immunology* (2003).

157. Guerra, C. *et al.* Control of Mycobacterium tuberculosis growth by activated natural killer cells. *Clinical and Experimental Immunology* **168,** 142–152 (Apr. 2012).

158. Portevin, D., Via, L. E., Eum, S. & Young, D. Natural killer cells are recruited during pulmonary tuberculosis and their ex vivo responses to mycobacteria vary between healthy human donors in association with KIR haplotype. *Cellular microbiology* **14,** 1734–1744 (Nov. 2012).

159. Coleman, C. M. & Wu, L. HIV interactions with monocytes and dendritic cells: viral latency and reservoirs. *Retrovirology* **6,** 51 (2009).

160. Laforge, M. *et al.* HIV/SIV infection primes monocytes and dendritic cells for apoptosis. *PLoS Pathogens* **7,** e1002087 (June 2011).

161. Hogg, A. *et al.* Activation of NK cell granulysin by mycobacteria and IL-15 is differentially affected by HIV. *Tuberculosis* **91 Suppl 1,** S75–81 (Dec. 2011).

162. Solis, M. *et al.* RIG-I-mediated antiviral signaling is inhibited in HIV-1 infection by a protease-mediated sequestration of RIG-I. *Journal of Virology* **85,** 1224–1236 (Feb. 2011).

163. Doehle, B. P., Hladik, F., McNevin, J. P., McElrath, M. J. & Gale, M. Human immunodeficiency virus type 1 mediates global disruption of innate antiviral signaling and immune defenses within infected cells. *Journal of Virology* **83,** 10395–10405 (Oct. 2009).

164. Russell, D. G. *et al. Mycobacterium tuberculosis* wears what it eats. *Cell host & microbe* **8,** 68–76 (July 2010).

165. Kim, M.-j. *et al.* Caseation of human tuberculosis granulomas correlates with elevated host lipid metabolism. *EMBO molecular medicine* **2,** 258–274 (July 2010).

166. Grotzke, J. E. *et al.* The *Mycobacterium tuberculosis* phagosome is a HLA-I processing competent organelle. *PLoS Pathogens* **5,** e1000374 (Apr. 2009).

Colophon

This document was typeset in 11pt Times Roman using LyX. Typesetting and layout utilised LaTeX 2e. Bibliographic data was stored in BibTeX format, and processed using *biber* and *biblatex*.

The pdf version has live hyperlinks for references and table of contents which may aid in reading.

Part VI

Appendix

A Code

A.1 Code for functions used throughout

A.1.1 Modified heatmap (heatmap_ad.R)

Changes made to original "heatmap.plus" command from heatmap.plus package:

Added options *noan* (number of annotations) and *rowlab* (row labels) to increase the width (*lhei*) of column annotation rows for the Colv matrix by the factor *noan*. This is useful because a large number of column annotations (e.g. based on phenodata) normally make the Colv matrix unreadable as it has a fixed width. The *rowlab* option switches the labelling of the column annotations on and off as desired. Credit for *noan* parametrisation of *lhei* is this post:

http://tolstoy.newcastle.edu.au/R/e9/help/10/02/6307.html.

```
#changelog?

heatmap_ad<-function (x, Rowv = NULL, Colv = if (symm) "Rowv" else NULL,
    distfun = dist, hclustfun = hclust, reorderfun = function(d,
        w) reorder(d, w), add.expr, symm = FALSE, revC = identical(Colv,
        "Rowv"), scale = c("row", "column", "none"), na.rm = TRUE,
    margins = c(5, 5), ColSideColors, RowSideColors, cexRow = 0.2 +
        1/log10(nr), cexCol = 0.2 + 1/log10(nc), labRow = NULL,
    labCol = NULL, main = NULL, xlab = NULL, ylab = NULL, keep.dendro =
    FALSE, noan=NULL,rowlab=TRUE,
    verbose = getOption("verbose"), ...)
{
    scale <- if (symm && missing(scale))
        "none"
    else match.arg(scale)
    if (length(di <- dim(x)) != 2 || !is.numeric(x))
        stop("'x' must be a numeric matrix")
    nr <- di[1]
    nc <- di[2]
    if (nr <= 1 || nc <= 1)
        stop("'x' must have at least 2 rows and 2 columns")
    if (!is.numeric(margins) || length(margins) != 2)
        stop("'margins' must be a numeric vector of length 2")
    doRdend <- !identical(Rowv, NA)
    doCdend <- !identical(Colv, NA)
    if (is.null(Rowv))
        Rowv <- rowMeans(x, na.rm = na.rm)
    if (is.null(Colv))
        Colv <- colMeans(x, na.rm = na.rm)
    if (doRdend) {
        if (inherits(Rowv, "dendrogram"))
```

```
            ddr <- Rowv
        else {
            hcr <- hclustfun(distfun(x))
            ddr <- as.dendrogram(hcr)
            if (!is.logical(Rowv) || Rowv)
                ddr <- reorderfun(ddr, Rowv)
        }
        if (nr != length(rowInd <- order.dendrogram(ddr)))
            stop("row dendrogram ordering gave index of wrong length")
    }
    else rowInd <- 1:nr
    if (doCdend) {
        if (inherits(Colv, "dendrogram"))
            ddc <- Colv
        else if (identical(Colv, "Rowv")) {
            if (nr != nc)
                stop("Colv = \"Rowv\" but nrow(x) != ncol(x)")
            ddc <- ddr
        }
        else {
            hcc <- hclustfun(distfun(if (symm)
                x
            else t(x)))
            ddc <- as.dendrogram(hcc)
            if (!is.logical(Colv) || Colv)
                ddc <- reorderfun(ddc, Colv)
        }
        if (nc != length(colInd <- order.dendrogram(ddc)))
            stop("column dendrogram ordering gave index of wrong length")
    }
    else colInd <- 1:nc
    x <- x[rowInd, colInd]
    labRow <- if(rowlab==TRUE)
      if (is.null(labRow))
        if (is.null(rownames(x)))
            (1:nr)[rowInd]

        else rownames(x)
    else labRow[rowInd]
    else NULL
    labCol <- if (is.null(labCol))
        if (is.null(colnames(x)))
            (1:nc)[colInd]
        else colnames(x)
    else labCol[colInd]
    if (scale == "row") {
        x <- sweep(x, 1, rowMeans(x, na.rm = na.rm))
        sx <- apply(x, 1, sd, na.rm = na.rm)
        x <- sweep(x, 1, sx, "/")
    }
    else if (scale == "column") {
        x <- sweep(x, 2, colMeans(x, na.rm = na.rm))
        sx <- apply(x, 2, sd, na.rm = na.rm)
        x <- sweep(x, 2, sx, "/")
    }

    lmat <- rbind(c(NA, 3), 2:1)
    lwid <- c(if (doRdend) 1 else 0.05, 4)
    lhei <- c((if (doCdend) 1 else 0.05) + if (!is.null(main)) 0.2 else 0,4)
```

```
    #lhei <- c((if (doCdend) 1 else 0.05) + if (!is.null(main)) 0.8 else
    0,4)
    if (!missing(ColSideColors)) {
        if (!is.matrix(ColSideColors))
            stop("'ColSideColors' must be a matrix")
        if (!is.character(ColSideColors) || dim(ColSideColors)[1] !=
            nc)
            stop("'ColSideColors' dim()[2] must be of length ncol(x)")
        lmat <- rbind(lmat[1, ] + 1, c(NA, 1), lmat[2, ] + 1)
         #lmat <- rbind(lmat[1, ] + 1, c(NA, 1), lmat[3, ] + 1)
        #lhei <- c(lhei[1], 0.2, lhei[2])
        lhei <- c(lhei[1], 0.1*noan, lhei[2])
    }
    if (!missing(RowSideColors)) {
        if (!is.matrix(RowSideColors))
            stop("'RowSideColors' must be a matrix")
        if (!is.character(RowSideColors) || dim(RowSideColors)[1] !=
            nr)
            stop("'RowSideColors' must be a character vector of length nrow(
    x)")
        lmat <- cbind(lmat[, 1] + 1, c(rep(NA, nrow(lmat) - 1),
            1), lmat[, 2] + 1)
        #lwid <- c(lwid[1], 0.2, lwid[2])
        lwid <- c(lwid[1], 0.2, lwid[2])
    }
    lmat[is.na(lmat)] <- 0
    if (verbose) {
        cat("layout: widths = ", lwid, ", heights = ", lhei,
            "; lmat=\n")
        print(lmat)
    }
    op <- par(no.readonly = TRUE)
    on.exit(par(op))
    layout(lmat, widths = lwid, heights = lhei, respect = FALSE)

    #1 draw row side col
    if (!missing(RowSideColors)) {
        par(mar = c(margins[1], 0, 0, 0.5))
        rsc = RowSideColors[rowInd, ]
        rsc.colors = matrix()
        rsc.names = names(table(rsc))
        rsc.i = 1
        for (rsc.name in rsc.names) {
            rsc.colors[rsc.i] = rsc.name
            rsc[rsc == rsc.name] = rsc.i
            rsc.i = rsc.i + 1
        }
        rsc = matrix(as.numeric(rsc), nrow = dim(rsc)[1])
        image(t(rsc), col = as.vector(rsc.colors), axes = FALSE)
        if (length(colnames(RowSideColors)) > 0) {
            axis(1, 0:(dim(rsc)[2] - 1)/(dim(rsc)[2] - 1), colnames(
    RowSideColors),
                las = 2, tick = FALSE)
        }
    }
    #2 draw col side col
    if (!missing(ColSideColors)) {
        par(mar = c(0.5, 0, 0, margins[2]))
        csc = ColSideColors[colInd, ]
```

```
146         csc.colors = matrix()
147         csc.names = names(table(csc))
148         csc.i = 1
149         for (csc.name in csc.names) {
150             csc.colors[csc.i] = csc.name
151             csc[csc == csc.name] = csc.i
152             csc.i = csc.i + 1
153         }
154         csc = matrix(as.numeric(csc), nrow = dim(csc)[1])
155         image(csc, col = as.vector(csc.colors), axes = FALSE)
156         if (length(colnames(ColSideColors)) > 0) {
157             axis(2, 0:(dim(csc)[2] - 1)/(dim(csc)[2] - 1), colnames(
    ColSideColors),
158                 las = 2, tick = FALSE)
159         }
160     }
161
162     #3 draw heatmap
163     par(mar = c(margins[1], 0, 0, margins[2]))
164     if (!symm || scale != "none") {
165         x <- t(x)
166     }
167     if (revC) {
168         iy <- nr:1
169         ddr <- rev(ddr)
170         x <- x[, iy]
171     }
172     else iy <- 1:nr
173
174     image(1:nc, 1:nr, x, xlim = 0.5 + c(0, nc), ylim = 0.5 +
175         c(0, nr), axes = FALSE, xlab = "", ylab = "", ...)
176     axis(1, 1:nc, labels = labCol, las = 2, line = -0.5, tick = 0,
177         cex.axis = cexCol)
178     if (!is.null(xlab))
179         mtext(xlab, side = 1, line = margins[1] - 1.25)
180     axis(4, iy, labels = labRow, las = 2, line = -0.5, tick = 0,
181         cex.axis = cexRow)
182     if (!is.null(ylab))
183         mtext(ylab, side = 4, line = margins[2] - 1.25)
184     if (!missing(add.expr))
185         eval(substitute(add.expr))
186     #4 draw row dendro
187     par(mar = c(margins[1], 0, 0, 0))
188     if (doRdend)
189         plot(ddr, horiz = TRUE, axes = FALSE, yaxs = "i", leaflab = "none")
190     else frame()
191
192     #5 draw col dendro
193     par(mar = c(0, 0, if (!is.null(main)) 2 else 0, margins[2]))#was1
194     if (doCdend)
195         plot(ddc, axes = FALSE, xaxs = "i", leaflab = "none")
196     else if (!is.null(main))
197         frame()
198     #6 title
199     if (!is.null(main))
200         title(main, cex.main = 1.3 * op[["cex.main"]])
201     invisible(list(rowInd = rowInd, colInd = colInd, Rowv = if (keep.dendro
    &&
202         doRdend) ddr, Colv = if (keep.dendro && doCdend) ddc))
```

```
203
204
205 }
```

Listing A.1: heatmapad.R

A.1.2 Comprehensive heatmap (1) (superHeatmap.R)

Convenient wrapper function for drawing heatmaps in the style used in the Berry et al paper [82]. This function adds options for correlation clustering and coloured dendrograms to the function parameters, and then draws the heatmap using *heatmap_ad.R*.

Correlation code is based on this tutorial on the ULB BIGRe lab site, and the coloured dendrogram solution is based on this post.

```
superHeatmap<-function(x,y=selected.probes,phenomatrix=phenomatrix,scale=
    scale,addTit=NULL){

#cluster
genes.cor <- cor(t(exprs(x)[y,]), use="pairwise.complete.obs",method="
    pearson")
genes.cor.dist <- as.dist(1-genes.cor)
genes.tree <- hclust(genes.cor.dist,method='average')
samples.cor.spearman <- cor(exprs(x)[y,],use="pairwise.complete.obs",method=
    "spearman")
samples.cor.spearman.dist <- as.dist(1-samples.cor.spearman)
samples.tree <- hclust(samples.cor.spearman.dist,method='average')
#dendrogram colors

#coloured dendrogram
smpCol <- as.dendrogram(samples.tree)
local({
  colLab <<- function(n) {
    if(is.leaf(n)) {
      a <- attributes(n)
      i <<- i+1
      #          attr(n, "nodePar") <-
      #              c(a$nodePar, list(lab.col = mycols[i], lab.font= i%%3),
    pch=NULL)
      attr(n, "edgePar") <-
        c(a$edgePar, list(col = mycols[i]),lwd=3)
    }
    n
  }
  mycols <- as.vector(phenomatrix[,1])[order.dendrogram(smpCol)]
  i <- 0
})
dL <- dendrapply(smpCol, colLab)

max.level <- max(abs(range(exprs(x)[y,])))

br <- c(seq(from=0,to=1,by=1/127),
            seq(from=1, to=5,by=4/126),
            max.level
)

#heatmap
heatmap_ad(as.matrix(exprs(x)[y,]),
           scale=scale, ## AVOID RESCALING THE VALUES OF ALL COLUMNS
           col=palette.BYR(),
           breaks=br,
           Rowv=as.dendrogram(genes.tree),
           Colv=dL,
           main=paste(nrow(exprs(x)[y,]),"probes",addTit),
           ColSideColors=phenomatrix,
```

```
47              margins=c(9,5),
48              #RowSideColors=rsc,
49              noan=2,labRow="",verbose=TRUE
50 )
51 }
```

Listing A.2: superHeatmap.R

A.1.3 Comprehensive heatmap (2) (superHeatmap.R)

Improves on superHeatmap.R by automating the selection of colour levels in the blue-yellow-red colour ramp.

```
superHeatmap2<-function(x,y=selected.probes,phenomatrix=phenomatrix,scale=
    scale,addTit=NULL,low=low,high=high){

#cluster
genes.cor <- cor(t(exprs(x)[y,]), use="pairwise.complete.obs",method="
    pearson")
genes.cor.dist <- as.dist(1-genes.cor)
genes.tree <- hclust(genes.cor.dist,method='average')
samples.cor.spearman <- cor(exprs(x)[y,],use="pairwise.complete.obs",method=
    "spearman")
samples.cor.spearman.dist <- as.dist(1-samples.cor.spearman)
samples.tree <- hclust(samples.cor.spearman.dist,method='average')
#dendrogram colors

#coloured dendrogram
smpCol <- as.dendrogram(samples.tree)
local({
  colLab <<- function(n) {
    if(is.leaf(n)) {
      a <- attributes(n)
      i <<- i+1
      #          attr(n, "nodePar") <-
      #              c(a$nodePar, list(lab.col = mycols[i], lab.font= i%%3),
    pch=NULL)
      attr(n, "edgePar") <-
        c(a$edgePar, list(col = mycols[i]),lwd=3)
    }
    n
  }
  mycols <- as.vector(phenomatrix[,1])[order.dendrogram(smpCol)]
  i <- 0
})
dL <- dendrapply(smpCol, colLab)

max.level <- max(abs(range(exprs(x)[y,])))
min.level <- min(abs(range(exprs(x)[y,])))
low<-1.05*min.level
high<-0.6*max.level

br <- c(seq(from=min.level,to=low,length.out=30),
        seq(from=low, to=high,length.out=105),
        seq(from=high,to=max.level,length.out=120),
        max.level
)

#heatmap
heatmap_ad(as.matrix(exprs(x)[y,]),
           scale=scale, ## AVOID RESCALING THE VALUES OF ALL COLUMNS
           col=palette.BYR(),
           #breaks=br,
           Rowv=as.dendrogram(genes.tree),
           Colv=dL,
           main=paste(nrow(exprs(x)[y,]),"probes",addTit),
           ColSideColors=phenomatrix,
```

```
51              margins=c(9,5),
52              #RowSideColors=rsc,
53              noan=2,labRow="",verbose=TRUE
54 )
55 }
```

Listing A.3: superHeatmap2.R

A.1.4 Modified Sample Relations plot (MDS) (plotSampleRelationsAD.R)

Modifies the function *plotSampleRelation* from the *lumi* package to plot points using specified plot character plotchar instead of the sample names; this makes the plot more readable. A background colour can also be specified.

```
plotSampleRelationsAD<-function (x, subset = NULL, cv.Th = 0.1, standardize = TRUE,
        method = c("cluster", "mds"), dimension = c(1, 2), color = NULL, backgr = NULL,plotchar=24,
        main = NULL, ...)
{
  if (is(x, "ExpressionSet")) {
    dataMatrix <- exprs(x)
  }
  else if (is.matrix(x)) {
    dataMatrix <- x
  }
  else {
    stop("The class of \"x\" should be matrix or LumiBatch!")
  }
  if (standardize)
    dataMatrix <- scale(dataMatrix)
  if (is.null(subset)) {
    probeList <- rownames(dataMatrix)
    if (is.null(probeList))
      probeList <- 1:nrow(dataMatrix)
    if (cv.Th > 0) {
      cv.gene <- apply(dataMatrix, 1, function(x) sd(x)/mean(x))
      subset <- probeList[abs(cv.gene) > cv.Th]
      if (is.null(main))
        main <- paste("Sample relations based on", length(subset),
                      "genes with sd/mean >", cv.Th)
    }
    else {
      subset <- probeList
      if (is.null(main))
        main <- paste("Sample relations based on", length(subset),
                      "genes")
    }
  }
  else {
    if (length(subset) == 1 && is.numeric(subset)) {
      subset <- sample(1:nrow(dataMatrix), min(subset,
                                               nrow(dataMatrix)))
    }
    if (is.null(main))
      main <- paste("Sample relations based on", length(subset),
                    "selected genes")
  }
  dd <- dist(t(dataMatrix[subset, ]))
  method <- match.arg(method)
  if (method == "cluster") {
    hc = hclust(dd, "ave")
    plot(hc, xlab = "Sample", main = main, ...)
    attr(hc, "geneNum") <- length(subset)
    attr(hc, "threshold") <- cv.Th
    return(invisible(hc))
  }
```

```
52   else {
53     mds.result <- cmdscale(dd, k = max(dimension), eig = TRUE)
54     ppoints <- mds.result$points
55     eig <- mds.result$eig
56     percent <- round(eig/sum(eig) * 100, 1)
57     if (is.null(color)) {
58       color <- 1
59     }
60     else {
61       if (!is.numeric(color)) {
62         allColor <- colors()
63         if (!all(is.element(color, allColor))) {
64           color <- as.numeric(factor(color, levels = unique(color)))
65         }
66       }
67     }
68     plot(ppoints[, dimension[1]], ppoints[, dimension[2]],
69          type = "n", xlab = paste("Principal Component ",
70                                   dimension[1], " (", percent[dimension[1]],
     "%)",
71                                   sep = ""), ylab = paste("Principal
     Component ",
72                                                           dimension[2], " ("
     , percent[dimension[2]], "%)",
73                                                           sep = ""), main =
     main, ...)
74     points(ppoints[, dimension[1]], ppoints[, dimension[2]],
75          col = color,bg=backgr, cex = 1,pch=plotchar)
76     attr(ppoints, "geneNum") <- length(subset)
77     attr(ppoints, "threshold") <- cv.Th
78     return(invisible(ppoints))
79   }
80 }
```

Listing A.4: plotSampleRelationsAD.R

A.1.5 Modified 3D Sample Relations plot

Modifies the function *plotSampleRelationsAD.R* to plot points in 3D space using the first three principal components. The plot3d command in this function requires the rgl library, which is loaded in the setup script.

```
plotSampleRelations3D_AD<-function (x, subset = NULL, cv.Th = 0.1,
    standardize = TRUE,
                                    method = c("cluster", "mds"), dimension = c
    (1, 2,3), color = NULL, backgr = NULL,plotchar=24,spheresize=2,
                                    main = NULL, ...)
{
  if (is(x, "ExpressionSet")) {
    dataMatrix <- exprs(x)
  }
  else if (is.matrix(x)) {
    dataMatrix <- x
  }
  else {
    stop("The class of \"x\" should be matrix or LumiBatch!")
  }
  if (standardize)
    dataMatrix <- scale(dataMatrix)
  if (is.null(subset)) {
    probeList <- rownames(dataMatrix)
    if (is.null(probeList))
      probeList <- 1:nrow(dataMatrix)
    if (cv.Th > 0) {
      cv.gene <- apply(dataMatrix, 1, function(x) sd(x)/mean(x))
      subset <- probeList[abs(cv.gene) > cv.Th]
      if (is.null(main))
        main <- paste("Sample relations based on", length(subset),
                      "genes with sd/mean >", cv.Th)
    }
    else {
      subset <- probeList
      if (is.null(main))
        main <- paste("Sample relations based on", length(subset),
                      "genes")
    }
  }
  else {
    if (length(subset) == 1 && is.numeric(subset)) {
      subset <- sample(1:nrow(dataMatrix), min(subset,
                                               nrow(dataMatrix)))
    }
    if (is.null(main))
      main <- paste("Sample relations based on", length(subset),
                    "selected genes")
  }
  dd <- dist(t(dataMatrix[subset, ]))
  method <- match.arg(method)
  if (method == "cluster") {
    hc = hclust(dd, "ave")
    plot(hc, xlab = "Sample", main = main, ...)
    attr(hc, "geneNum") <- length(subset)
    attr(hc, "threshold") <- cv.Th
    return(invisible(hc))
  }
```

```
52   else {
53     mds.result <- cmdscale(dd, k = max(dimension), eig = TRUE)
54     ppoints <- mds.result$points
55     eig <- mds.result$eig
56     percent <- round(eig/sum(eig) * 100, 1)
57     if (is.null(color)) {
58       color <- 1
59     }
60     else {
61       if (!is.numeric(color)) {
62         allColor <- colors()
63         if (!all(is.element(color, allColor))) {
64           color <- as.numeric(factor(color, levels = unique(color)))
65         }
66       }
67     }
68     plot3d(ppoints[, dimension[1]], ppoints[, dimension[2]], ppoints[,
     dimension[3]],
69          type = "s", size=spheresize,
70            xlab = paste("Principal Component ",dimension[1], " (", percent[
     dimension[1]], "%)",sep = ""),
71            ylab = paste("Principal Component ",dimension[2], " (", percent[
     dimension[2]], "%)",sep = ""),
72            zlab = paste("Principal Component ",dimension[3], " (", percent[
     dimension[3]], "%)",sep = ""),
73            main = main,col=color)
74     #points(ppoints[, dimension[1]], ppoints[, dimension[2]], ppoints[,
     dimension[3]],col = color,bg=backgr, cex = 1,pch=plotchar)
75
76   }
77 }
```

Listing A.5: plotSampleRelations3DAD.R

A.2 System setup

```
#metadata####
#IMPI_MA microarray analysis pipeline
#Universal System Setup
#Armin Deffur
#File created 18-02-2013
#Version 1.0

#1. System Setup####

#initialise session
rm(list=ls())
graphics.off()

#Local folder configuration####

#Projects root (scripts and output)
dir.root <- '/Users/armindeffur/Documents/002_Science_universal/Active_
    Research/IMPI-Microarray/Projects'

#Functions

#R-scripts root
dir.R.files <- file.path(dir.root, "01_scripts/06_IMPI-MA/Analysis/03_
    Functions")
print(paste("R scripts source", dir.R.files))
## R utilities folder
dir.util <- file.path(dir.R.files, 'util')

#Pipelines

#Data root
dir.data_root<-'/Users/armindeffur/Documents/002_Science_universal/Active_
    Research/IMPI-Microarray/Data'

#Load utilities####
source(file.path(dir.util, 'util.R'))
source(file.path(dir.util, 'util_chip_analysis.R'))
source(file.path(dir.R.files,"plotSampleRelationsAD.R"))
source(file.path(dir.R.files,"plotSampleRelations3D_AD.R"))
source(file.path(dir.R.files,"heatmap_ad.R"))
source(file.path(dir.R.files,"heatmap_ad2.R"))

#Global parameters####
verbosity <- 1
#export.formats.plots <- c("eps","pdf", "png","jpg")
#export.formats.plots <- c("pdf", "png","jpg")
export.formats.plots <- c("pdf")
export.formats.obj <- c("table")
#height=16#use for printing?
height=8#use for presentation graphics
par(mfrow=c(1,1),mar=c(5, 4, 4, 2) + 0.1,xaxt="s")
parbackup<-par(mfrow=c(1,1),mar=c(5, 4, 4, 2) + 0.1,xaxt="s",cex.main=1)
plotnumber<-1

## Specify if drawings should be done in colors or not.
in.colors <- T
```

```
54
55 #load all libraries
56 library(lumi)
57 library(annotate)
58 library(lumiHumanAll.db)
59 library(lumiHumanIDMapping)
60 library(rgl)
61 library(limma)
62 library(xtable)
63 library(gplots)
64 library(VennDiagram)
65 library(tools)
66 library(vegan)
67 library(CellMix)
68 library(csSAM)
69 library(RColorBrewer)
70 library(pwr)
71 library(ClassDiscovery)
72 library(WGCNA)
73 library(RColorBrewer)
74 library(beadarray)
75 library(illuminaHumanv3.db)
76 library(illuminaHumanv4.db)
```

Listing A.6: SystemSetup.R

A.3 Data manager (general)

```
#metadata####
#IMPI_MA microarray analysis pipeline
#Universal Data Manager
#Armin Deffur
#File created 18-02-2013
#Version 1.0

#System_setup.R####

source("/Users/armindeffur/Documents/002_Science_universal/Active_Research/
    IMPI-Microarray/Projects/01_scripts/06_IMPI-MA/Analysis/00_System/System
    _setup.R")

#Local Setup####

## Individual data folders (specific for each part)
dir.gx <- file.path(dir.data_root, '09_IMPI_MA_DATA/04_GX_format')
print(paste("GX12-format data repository", dir.gx))
dir.gs <- file.path(dir.data_root, '09_IMPI_MA_DATA_unversioned/05_
    GenomeStudio_reports')
print(paste("GenomeStudio report data repository", dir.gs))
dir.bgx <- file.path(dir.data_root, '09_IMPI_MA_DATA_unversioned/06_Illumina
     Manifest files/HT12v4')
print(paste("Illumina manifest file repository", dir.gx))
dir.jpg <- file.path(dir.data_root, '09_IMPI_MA_DATA_unversioned/07_
    imageData/JPG')
print(paste("Raw image data (.jpg) repository", dir.jpg))
dir.tif <- file.path(dir.data_root, '09_IMPI_MA_DATA_unversioned/07_
    imageData/TIFF')
print(paste("Raw image data (.tif) repository", dir.tif))
dir.pheno <- file.path(dir.data_root, '09_IMPI_MA_DATA/08_PhenoData')
print(paste("Phenodata data repository", dir.pheno))

#Output folders
dir.home<-dir.root
dir.main <- file.path(dir.root, '02_output/06_IMPI-MA','AllData')

#Current date string used for versioning output
ds<-Sys.time()

#versioned output
dir.output.version<-file.path(dir.main,paste("version_",ds))

## Define folder for storing results NB do this for each analysis
dir.results <- file.path(dir.output.version, "results")
print(paste("Results will be saved to", dir.results))

## Define folder for saving figures
dir.figures <- file.path(dir.output.version, "figures")
print(paste("Figures will be saved to", dir.figures))

## Define folder for saving RData files
dir.rdata <- file.path(dir.data_root,'09_IMPI_MA_DATA/05_RData')
print(paste("Data will be saved to", dir.rdata))

for (dir in c(dir.main, dir.figures, dir.results)) {
```

```
  if (!file.exists(dir)) {
    dir.create(dir, recursive=T,showWarnings=T)
  }
}

figure<-1

#WD####
setwd(dir.main)

#Data for Differential Expression

#Data for Deconvolution

#Data for WGCNA

#Import expression and phenotype data using lumi####

fileName<-file.path(dir.gx,"Armin_Hu_17_8_12_No uRNA_bkg sub_NO Norm__Sample_Probe_Profile.txt")
fileNameAnno<-file.path(dir.gs,"Armin_Hu_17_8_12_No uRNA_bkg sub_NO Norm_FinalReport.txt")
bgxfile<-file.path(dir.bgx,"HumanHT-12_V4_0_R1_15002873_B.bgx")

#PHENODATA ORIGINAL
#the phenodata below assigns TB status according to Pericardial culture. This misclassifies TB-PC suspects who were culture positive elsewhere as "probable" TB (wrong)
#phenodataPath<-file.path(dir.pheno,"phenoGS_edited_csv_colour_25_9_2012.csv")

#PHENODATA EDIT 1
#the phenodata (class3 variable) below defines active TB as ANY positive site. This means that some probable TBPC cases are defnite TB cases. This is relevant as the !blood! signature will be determined by any (??) TB site. This needs to be tested where fluid is negative but another site positive. It is unclear if this applies to TBM, though. (issues of blood-brain-barrier, etc.)
#Important: when changing this, factors of diagnostic class may change, depending which "probable" case is made "definite".
#phenodataPath<-file.path(dir.pheno,"phenoGS_edited_csv_colour_28_10_2012.csv")

#PHENODATA EDIT 2
#With this in mind, the phenodata source file has been updated to change sample 20868control back to probableActiveTB, as the only positive culture was CSF. It seems unwise to include only one case of TBM, while there are many TB-PC and PTB.
#the phenodata file was edited: sample PCA141 blood and fluid assignments to chips have been exchanged, as earlier mds plots strongly suggested this. All results may change that depend on the correct assignment.

#IMPI-MA data
phenodataPath<-file.path(dir.pheno,"phenoGS_edited_csv_colour_17_01_2013.csv")
sampleInfoDF<-read.csv(phenodataPath, header = TRUE, colClasses = "character", comment.char = "",check.names = FALSE)
allArrays<-lumiR.batch(fileName,lib.mapping='lumiHumanIDMapping',sampleInfoFile=sampleInfoDF,verbose=TRUE)
```

```
90 colnames(exprs(allArrays))<-allArrays$sample_name
91
92 #Berry data
93 berry.sampleInfoDF.train<-read.csv("/Users/armindeffur/Documents/002_Science
       _universal/Active_Research/IMPI-Microarray/Data/03_TB_AOG/data/Berry_
       pheno/phenoData_train.csv", header = TRUE, colClasses = "character",
       comment.char = "",check.names = FALSE)
94 berry.train<-lumiR.batch("/Users/armindeffur/Documents/002_Science_universal
       /Active_Research/IMPI-Microarray/Data/03_TB_AOG/New_data/MatthewRequest_
       02Feb11_TrainingSet.txt",lib.mapping='lumiHumanIDMapping',sampleInfoFile
       =berry.sampleInfoDF.train,verbose=TRUE)
95
96 berry.sampleInfoDF.test<-read.csv("/Users/armindeffur/Documents/002_Science_
       universal/Active_Research/IMPI-Microarray/Data/03_TB_AOG/data/Berry_
       pheno/phenoData_test.csv", header = TRUE, colClasses = "character",
       comment.char = "",check.names = FALSE)
97 berry.test<-lumiR.batch("/Users/armindeffur/Documents/002_Science_universal/
       Active_Research/IMPI-Microarray/Data/03_TB_AOG/New_data/MatthewRequest_
       02Feb11_TestSet.txt",lib.mapping='lumiHumanIDMapping',sampleInfoFile=
       berry.sampleInfoDF.test,verbose=TRUE)
98
99 berry.sampleInfoDF.val<-read.csv("/Users/armindeffur/Documents/002_Science_
       universal/Active_Research/IMPI-Microarray/Data/03_TB_AOG/data/Berry_
       pheno/phenoData_val.csv", header = TRUE, colClasses = "character",
       comment.char = "",check.names = FALSE)
100 berry.val<-lumiR.batch("/Users/armindeffur/Documents/002_Science_universal/
       Active_Research/IMPI-Microarray/Data/03_TB_AOG/New_data/MatthewRequest_
       02Feb11_ValidationSet.txt",lib.mapping='lumiHumanIDMapping',
       sampleInfoFile=berry.sampleInfoDF.val,verbose=TRUE)
101
102 #subset allArrays by compartment
103 fluidArrays<-allArrays[,allArrays$Compartment=="fluid"]
104 bloodArrays<-allArrays[,allArrays$Compartment=="blood"]
105
106 #Data subsetting####
107
108 #make subsets
109
110 #CSC scheme: compartment, subset, contrast
111
112 save(berry.train,file=file.path(dir.rdata,'berry_train.RData'))
113 save(berry.test,file=file.path(dir.rdata,'berry_test.RData'))
114 save(berry.val,file=file.path(dir.rdata,'berry_val.RData'))
115
116 save(allArrays,file=file.path(dir.rdata,'IMPI_MA.RData'))
117
118 hivNegBloodArrays<-bloodArrays[,bloodArrays$HIV.Status=="negative"] #tb
119 B.N.AX<-hivNegBloodArrays[,hivNegBloodArrays$class3=="activeTB"|
       hivNegBloodArrays$class3=="notActiveTB"]#training
120 B.N.aX<-hivNegBloodArrays[,hivNegBloodArrays$class3=="probableActiveTB"|
       hivNegBloodArrays$class3=="notActiveTB"]#validation
121 save(B.N.AX,file=file.path(dir.rdata,'blood_hivNeg_aTB_noTB.RData'))
122 save(B.N.aX,file=file.path(dir.rdata,'blood_hivNeg_probTB_noTB.RData'))
123
124 hivPosBloodArrays<-bloodArrays[,bloodArrays$HIV.Status=="positive"]
125 B.P.AX<-hivPosBloodArrays[,hivPosBloodArrays$class3=="activeTB"|
       hivPosBloodArrays$class3=="notActiveTB"]
126 B.P.aX<-hivPosBloodArrays[,hivPosBloodArrays$class3=="probableActiveTB"|
       hivPosBloodArrays$class3=="notActiveTB"]
```

```
127 save(B.P.AX,file=file.path(dir.rdata,'blood_hivPos_aTB_noTB.RData'))
128 save(B.P.aX,file=file.path(dir.rdata,'blood_hivPos_probTB_noTB.RData'))
129
130 ##
131 B.aTB.PN<-bloodArrays[,bloodArrays$class3=="activeTB"] #tb
132 save(B.aTB.PN,file=file.path(dir.rdata,'blood_aTB_HIVposNeg.RData'))
133 B.noTB.PN<-bloodArrays[,bloodArrays$class3=="notActiveTB"] #tb
134 save(B.noTB.PN,file=file.path(dir.rdata,'blood_noTB_HIVposNeg.RData'))
135 ##
136
137 B.C<-bloodArrays[,bloodArrays$type=="PC"] #hiv,hd
138 PF.C<-fluidArrays[,fluidArrays$type=="PC"] #hiv, hd
139 save(B.C,file=file.path(dir.rdata,'blood_TBPC.RData'))
140 save(PF.C,file=file.path(dir.rdata,'fluid_TBPC.RData'))
141
142 C.BPF<-allArrays[,allArrays$matching=="m"] #comp
143 C.N.BPF<-C.BPF[,C.BPF$HIV.Status=="negative"]
144 C.P.BPF<-C.BPF[,C.BPF$HIV.Status=="positive"]
145 save(C.BPF,file=file.path(dir.rdata,'TBPC_matchedBloodFluid.RData'))
146 save(C.N.BPF,file=file.path(dir.rdata,'TBPC_hivNeg_matchedBloodFluid.RData')
    )
147 save(C.P.BPF,file=file.path(dir.rdata,'TBPC_hivPos_matchedBloodFluid.RData')
    )
148
149 datasets<-list(
150   "Berry traininig set"=berry.train,
151   "Berry test set"=berry.test,
152   "Berry validation set"=berry.val,
153   "IMPI_MA"=allArrays,
154   "Blood, HIV neg, active or not active TB"=B.N.AX,
155   "Blood, HIV neg, probable or not active TB"=B.N.aX,
156   "Blood, HIV pos, active or not active TB"=B.P.AX,
157   "Blood, HIV pos, probable or not active TB"=B.P.aX,
158   "Blood, all TBPC"=B.C,
159   "Pericardial Fluid, all TBPC"=PF.C,
160   "Matched blood and pericardial fluid"=C.BPF,
161   "Matched blood and pericardial fluid, HIV neg"=C.N.BPF,
162   "Matched blood and pericardial fluid, HIV pos"=C.P.BPF)
163
164 #Berry signatures
165 #file=file.path(dir.gx,"Nature_paper_transcripts","P22_42_04Mar09_Training_
    393_GX11.txt")
166 #np393<-read.table(file,header=TRUE,sep="\t")
167 #char393<-as.character(np393[,1])
168
169 #berry.sig<-char393
170 #save(berry.sig,file=file.path(dir.rdata,'berry.sig.RData'))
171
172 #need to get 86 transcript list; downloaded as xls
173
174 #import
175 file86=file.path(dir.gx,"Nature_paper_transcripts","nature09247-s4.csv")
176 np86<-read.csv(file86,header=TRUE,sep=",")
177 berry.86<-np86
178 save(berry.86,file=file.path(dir.rdata,'berry.86.RData'))
179
180 #import manual berry393
181
182 #not done, need to fix format...
```

```
183 file393=file.path(dir.gx,"Nature_paper_transcripts","nature09247-s2.csv")
184 np393<-read.csv(file393,header=TRUE,sep=",")
185 berry.393<-np393
186 save(berry.393,file=file.path(dir.rdata,'berry.393.RData'))
187
188
189 #convert illumina ID to nuID
190 #as character
191 #save as .RData
```

Listing A.7: Data-manager.R

A.4 RT-PCR

```
1
2  #setup####
3  source("/Users/armindeffur/Documents/002_Science_universal/Active_Research/
       IMPI-Microarray/Projects/01_scripts/03_RT-PCR/11_RTPCR_paper/RTPCR_
       System_setup.R")
4
5  par("xpd"=FALSE)
6  ## Individual data folders (specific for each part)
7  dir.rtpcrdata <- file.path(dir.data_root,'02_IMPI_immunology data/Assays/RT-
       PCR')
8
9  dir.phenodata <- file.path("/Users/armindeffur/Documents/002_Science_
       universal/Active_Research/IMPI-Microarray/Projects/02_output/02_
       Phenotype exploration/")
10
11 #Output folders
12 dir.home<-dir.root
13 dir.main <- file.path(dir.root, '02_output/03_RT-PCR/11_RTPCR_paper')
14
15 #Current date string used for versioning output
16 ds<-Sys.time()
17
18 #versioned output
19 dir.output.version<-file.path(dir.main,paste("ver_",ds))
20
21 ## Define folder for storing results NB do this for each analysis
22 dir.results <- file.path(dir.output.version, "results")
23 ## Define folder for saving figures
24 dir.figures <- file.path(dir.output.version, "figures")
25 for (dir in c(dir.main, dir.figures, dir.results)) {
26   if (!file.exists(dir)) {
27     dir.create(dir, recursive=T,showWarnings=T)
28   }
29 }
30
31 sink(type="output",file=file.path(dir.results,"session_output.txt"))
32 sessionInfo()
33 print(paste("Begin time:",ds))
34 print(paste("Results will be saved to", dir.results))
35 print(paste("Figures will be saved to", dir.figures))
36
37
38 height=6
39 #additional functions####
40 redgreen<-function(n){c(hsv(h=2/6,v=seq(1,0,length=n/2)),hsv(h=0/6,v=seq
       (0,1,length=n/2)))}
41 #redgreen<-colorRampPalette(c("green","lightgreen","white","yellow","orange
       ","red","purple")) ## (n)
42 #redgreen<-colorRampPalette(c("lightblue","gray","white","yellow","red","
       purple")) ## (n)
43 #redgreen<-colorRampPalette(c("lightgreen","lightgray","white","red")) ## (n
       )
44 redgreen<-colorRampPalette(c("purple","blue","lightblue","lightgreen","
       cornsilk","wheat1","red")) ## (n)
45 bluered<-colorRampPalette(c("blue","lightblue","white","pink","red")) ## (n)
46
```

```
interleave <- function(v1,v2)
{
ord1 <- 2*(1:length(v1))-1
ord2 <- 2*(1:length(v2))
c(v1,v2)[order(c(ord1,ord2))]
}
#Data####
#delta CT values
data2<-read.xls(file.path(dir.rtpcrdata,"rtDat2.XLS"))
#ELISA and luminex
data2.prot<-read.xls(file.path(dir.rtpcrdata,"elDat.XLS"))
#Phenotype information
#make sure to use the latest! The below can vary as the output from the phenotype explorer in Mathematica changes!
phenodata<-read.xls(file.path(dir.phenodata,"SET1_RT-PCRsumdata + class20130409.xls"))
#Combine two datasets
data3<-merge(phenodata,data2,by.x="ID_KERRYN",by.y="ID")
data3.prot<-merge(phenodata,data2.prot,by.x="ID_KERRYN",by.y="PersonID")
#About the patients: Table 1 calculations####

#this stuff needs to be moved into a df
pts2<-phenodata
#variables
hiv<-pts2$HIV
table(hiv)

tbs<-pts2$TB.class
tbs2<-ftable(tbs,hiv)
tbs2
tbs.p<-chisq.test(tbs2)$p.value

sex<-pts2$SEX
table(sex)
sex2<-ftable(sex,hiv)
sex2
sex.p<-chisq.test(sex2)$p.value

fluid<-pts2$Fluid
table(fluid)
fluid2<-ftable(fluid,hiv)
fluid2
fluid2.p<-chisq.test(fluid2)$p.value

onart=pts2$ARVS
table(onart)
onart2<-ftable(onart,hiv)
onart2
onart2.p<-chisq.test(onart2)

onster=pts2$STEROIDS
table(onster)
onster2<-ftable(onster,hiv)
onster2
onster.p<-chisq.test(onster2)

m<-tapply(pts2$AGE_CALC,hiv,median,na.rm=T)
i<-tapply(pts2$AGE_CALC,hiv,summary)
n<-tapply(pts2$AGE_CALC,hiv,length)
```

```
summary(pts2$AGE_CALC)
#boxplot(pts2$AGE_CALC)
cbind(median=m,summary=as.character(i),n=n)
wilcox.test(pts2$AGE_CALC~hiv)

m<-tapply(pts2$CD4,hiv,median,na.rm=T)
i<-tapply(pts2$CD4,hiv,summary)
n<-tapply(pts2$CD4,hiv,length)
summary(pts2$CD4)
cbind(median=m,summary=as.character(i),n=n)
wilcox.test(pts2$CD4~hiv)

m<-tapply(pts2$ADA,hiv,median,na.rm=T)
i<-tapply(pts2$ADA,hiv,summary)
n<-tapply(pts2$ADA,hiv,length)
summary(pts2$ADA)
cbind(median=m,summary=as.character(i),n=n)
wilcox.test(pts2$ADA~hiv)

m<-tapply(pts2$IFN.gamma,hiv,median,na.rm=T)
i<-tapply(pts2$IFN.gamma,hiv,summary)
n<-tapply(pts2$IFN.gamma,hiv,length)
summary(pts2$IFN.gamma)
cbind(median=m,summary=as.character(i),n=n)
wilcox.test(pts2$IFN.gamma~hiv)

#Select compartment####

#Blood only
data3blood<-data3[data3$compart=="Blood",]
#Fluid only
data3fluid<-data3[data3$compart=="Fluid",]
#the phenotype categories####
compart<-as.factor(data3$compart);levels(compart)<-c("red","yellow")
sex<-as.factor(data3$SEX);levels(sex)<-c("white","blue","pink")
h<-as.factor(data3$HIV);levels(h)<-c("green","orange","white")
lowcd<-as.factor(data3$CD4.200);levels(lowcd)<-c("white","black","gray")
cd4<-as.factor(data3$CD4);levels(cd4)<-sequential_hcl(28, h=100,c = 100,
    power = 1.2)
hd<-as.factor(data3$ECP);levels(hd)<-brewer.pal(3,"Set1")
tbs<-as.factor(data3$TB.PC);levels(tbs)<-brewer.pal(3,"Set2")
tbc<-as.factor(data3$PCF.TB.culture);levels(tbc)<-brewer.pal(4,"Set3")
fluid<-as.factor(data3$Fluid);levels(fluid)<-c("white","white","red","pink",
    "yellow")
#phenomatrix definitions
#phenomatrix<-matrix(cbind(as.character(tbc),as.character(tbs),as.character(
    hd),as.character(h),as.character(lowcd),as.character(cd4),as.character(
    sex),as.character(compart),as.character(fluid)),ncol=9)
#colnames(phenomatrix)<-c("TBC","TBS","HD","HIV","lowCD4","CD4","SEX","COMP
    ","Fluid")
phenomatrix<-matrix(cbind(as.character(h),as.character(compart)),ncol=2)
colnames(phenomatrix)<-c("HIV","COMP")

#for blood
compart.blood<-as.factor(data3blood$compart);levels(compart.blood)<-c("red",
    "yellow")
sex.blood<-as.factor(data3blood$SEX);levels(sex.blood)<-c("white","blue","
    pink")
```

```
155 h.blood<-as.factor(data3blood$HIV);levels(h.blood)<-c("green","orange","
        white")
156 lowcd.blood<-as.factor(data3blood$CD4<200);levels(lowcd.blood)<-c("white","
        black")
157 cd4.blood<-as.factor(data3blood$CD4);levels(cd4.blood)<-sequential_hcl(28, h
        =100,c = 100, power = 1.2)
158 hd.blood<-as.factor(data3blood$ECP);levels(hd.blood)<-brewer.pal(3,"Set1")
159 tbs.blood<-as.factor(data3blood$TB.PC);levels(tbs.blood)<-brewer.pal(3,"Set2
        ")
160 tbc.blood<-as.factor(data3blood$PCF.TB.culture);levels(tbc.blood)<-brewer.
        pal(4,"Set3")
161 fluid.blood<-as.factor(data3blood$Fluid);levels(fluid.blood)<-c("white","
        white","red","pink","yellow")
162 phenomatrix.blood<-matrix(cbind(as.character(h.blood),as.character(lowcd.
        blood)),ncol=2)
163 colnames(phenomatrix.blood)<-c("HIV","low CD4")
164
165 #for fluid
166 compart.fluid<-as.factor(data3fluid$compart);levels(compart.fluid)<-c("red",
        "yellow")
167 sex.fluid<-as.factor(data3fluid$SEX);levels(sex.fluid)<-c("white","blue","
        pink")
168 h.fluid<-as.factor(data3fluid$HIV);levels(h.fluid)<-c("green","orange","
        white")
169 lowcd.fluid<-as.factor(data3fluid$CD4<200);levels(lowcd.fluid)<-c("white","
        black")
170 cd4.fluid<-as.factor(data3fluid$CD4);levels(cd4.fluid)<-sequential_hcl(28, h
        =100,c = 100, power = 1.2)
171 hd.fluid<-as.factor(data3fluid$ECP);levels(hd.fluid)<-brewer.pal(3,"Set1")
172 tbs.fluid<-as.factor(data3fluid$TB.PC);levels(tbs.fluid)<-brewer.pal(3,"Set2
        ")
173 tbc.fluid<-as.factor(data3fluid$PCF.TB.culture);levels(tbc.fluid)<-brewer.
        pal(4,"Set3")
174 fluid.fluid<-as.factor(data3fluid$Fluid);levels(fluid.fluid)<-c("white","
        white","red","pink","yellow")
175 phenomatrix.fluid<-matrix(cbind(as.character(h.fluid),as.character(lowcd.
        fluid)),ncol=2)
176 colnames(phenomatrix.fluid)<-c("HIV","low CD4")
177 #Unfiltered  data####
178 #datamat<-as.matrix(data3[,-40:-1])
179 #rownames(datamat)<-data3$ID2
180 #transform RT-PCR data to LFC
181 #datamat=log10(2^-datamat)
182 #Normalization methods####
183 #datamat<-t(normalizeQuantileRank.matrix(t(datamat),robust=TRUE))
184 #datamat<-t(normalizeQuantile(t(datamat)))
185 #datamat<-t(normalizeBetweenArrays(t(datamat)))
186 #datamat<-t(normalizeMedianValues(t(datamat)))
187 #datamat<-t(normalizeAverage.matrix(t(datamat)))
188
189 # #Start here: separate quantile normalization of blood and fluid, then
        combining to one matrix which is median scaled, then transformed to
        expression levels####
190 bloodmat<-as.matrix(data3blood[,-40:-1])
191 rownames(bloodmat)<-data3blood$ID2
192 # bloodmatN<-t(normalizeQuantileRank.matrix(t(bloodmat),robust=TRUE))
193 # #bloodmatN<-t(normalizeBetweenArrays(t(bloodmat),method="quantile"))
194 # #bloodmatN=log10(2^-bloodmatN)
195 #
```

```
fluidmat<-as.matrix(data3fluid[,-40:-1])
rownames(fluidmat)<-data3fluid$ID2
# fluidmatN<-t(normalizeQuantileRank.matrix(t(fluidmat),robust=TRUE))
# #fluidmatN<-t(normalizeBetweenArrays(t(fluidmat),method="quantile"))
# #fluidmatN=log10(2^-fluidmatN)
#
# datamat<-rbind(bloodmatN,fluidmatN)
# datamat<-t(normalizeMedianValues(t(datamat)))

#not normalized at all (argument for this approach is that all values are normalized to beta Actin already, which is universally constant):
datamat<-rbind(bloodmat,fluidmat)

#transformation
datamat=log10(2^-datamat)

ref.min<-min(datamat)
ref.max<-max(datamat)

#phenomatrix####
phenomatrix=cbind(c(rep("red",27),rep("yellow",27)),c(rep("red",27),rep("yellow",27)))
calphenomatrix=rbind(c("blue","blue"),c("blue","blue"),phenomatrix)
#gene categories
cv=c(rep("red",8),rep("green",2),rep("yellow",3),rep("blue",3),rep("white",5),rep("purple",3),rep("pink",7),rep("gray",11))
genecol=matrix(c(cv,cv),ncol=2)
rownames(genecol)<-colnames(datamat)

##Unfiltered raw data: Boxplots####

figure<-1
setwd(dir.figures)

par(parbackup)
boxplot(datamat~factor(c(rep("blood",nrow(data3)/2),rep("fluid",nrow(data3)/2))),col=c("red","yellow"),main="Overall expression values for blood and pericardial fluid")
export.plot(file.prefix=paste("fig",figure,".","boxplot overall expression blood and fluid"),export.formats=export.formats.plots,height=height*1.5,width=(1 + sqrt(5))/2*height*1.5)
figure<-figure+1

m.blood<-median(datamat[1:27,])
m.fluid<-median(datamat[28:54,])
blood.fluid.overall.comparison<-wilcox.test(datamat~factor(c(rep("blood",nrow(data3)/2),rep("fluid",nrow(data3)/2))))

boxplot(t(datamat),las=2,col=c(rep("red",nrow(data3)/2),rep("yellow",nrow(data3)/2)),main="Boxplot of expression values by sample")
export.plot(file.prefix=paste("fig",figure,".","boxplot overall expression blood and fluid"),export.formats=export.formats.plots,height=height*1.5,width=(1 + sqrt(5))/2*height*1.5)
figure<-figure+1

npar<-par(mar=c(6,3,3,19),xpd=TRUE)

boxplot(datamat,las=2,col=cv,main="Expression in blood and fluid")
```

```
243 legend(45,-5,c("TH1","TH2","TH17","Immunoregulation","Neutrophil","Growth
        factors","Fibrosis","MMPs"),fill=c("red","green","yellow","blue","white"
        ,"purple","pink","gray"),cex=.8)
244 export.plot(file.prefix=paste("fig",figure,".","boxplot"),export.formats=
        export.formats.plots,height=height*1.5,width=(1 + sqrt(5))/2*height*1.5)
245 figure<-figure+1
246
247 boxplot(datamat[1:27,],las=2,col=cv,main="Expression in blood")
248 legend(45,-5,c("TH1","TH2","TH17","Immunoregulation","Neutrophil","Growth
        factors","Fibrosis","MMPs"),fill=c("red","green","yellow","blue","white"
        ,"purple","pink","gray"),cex=.8)
249 export.plot(file.prefix=paste("fig",figure,".","boxplot"),export.formats=
        export.formats.plots,height=height*1.5,width=(1 + sqrt(5))/2*height*1.5)
250 figure<-figure+1
251
252 boxplot(datamat[28:54,],las=2,col=cv,main="Expression in fluid")
253 legend(45,-5,c("TH1","TH2","TH17","Immunoregulation","Neutrophil","Growth
        factors","Fibrosis","MMPs"),fill=c("red","green","yellow","blue","white"
        ,"purple","pink","gray"),cex=.8)
254 export.plot(file.prefix=paste("fig",figure,".","boxplot"),export.formats=
        export.formats.plots,height=height*1.5,width=(1 + sqrt(5))/2*height*1.5)
255 figure<-figure+1
256
257 ord<-interleave(seq(1,42),seq(43,84))
258 newdat<-cbind(datamat[1:27,],datamat[28:54,])
259 boxplot(newdat[,ord],col=c("red","yellow"),las=2,main="Expression levels of
        all genes")
260 legend(89,-5,c("blood","fluid"),fill=c("red","yellow"),cex=.8)
261 export.plot(file.prefix=paste("fig",figure,".","boxplot"),export.formats=
        export.formats.plots,height=height*1.5,width=(1 + sqrt(5))/2*height*1.5)
262 figure<-figure+1
263
264 npar<-par(mar=c(5,3,3,3),xpd=TRUE,mfrow=c(2,2))
265
266 boxplot(newdat[,ord][,1:16],col=c("red","yellow"),ylab="expression",las=2,
        main="TH1")
267 boxplot(newdat[,ord][,17:20],col=c("red","yellow"),ylab="expression",las=2,
        main="TH2")
268 boxplot(newdat[,ord][,21:26],col=c("red","yellow"),ylab="expression",las=2,
        main="TH17")
269 boxplot(newdat[,ord][,27:32],col=c("red","yellow"),ylab="expression",las=2,
        main="Immunoregulation")
270 #legend("bottomleft",c("blood","fluid"),fill=c("red","yellow"),cex=.8,horiz=
        T)
271 export.plot(file.prefix=paste("fig",figure,".","boxplot"),export.formats=
        export.formats.plots,height=height*1.5,width=(1 + sqrt(5))/2*height*1.5)
272 figure<-figure+1
273
274 boxplot(newdat[,ord][,33:42],col=c("red","yellow"),ylab="expression",las=2,
        main="Neutrophil")
275 boxplot(newdat[,ord][,43:48],col=c("red","yellow"),ylab="expression",las=2,
        main="Growth factors")
276 boxplot(newdat[,ord][,49:62],col=c("red","yellow"),ylab="expression",las=2,
        main="Fibrosis")
277 boxplot(newdat[,ord][,63:84],col=c("red","yellow"),ylab="expression",las=2,
        main="MMPs")
278 #legend(15.5,-10,c("blood","fluid"),fill=c("red","yellow"))par(parbackup)
279 export.plot(file.prefix=paste("fig",figure,".","boxplot"),export.formats=
        export.formats.plots,height=height*1.5,width=(1 + sqrt(5))/2*height*1.5)
```

```
280
281 figure<-figure+1
282 par(parbackup)
283
284
285 npar<-par(mar=c(5.5,3,3,3),xpd=TRUE,mfrow=c(4,2))
286
287 boxplot(newdat[,ord][,1:16],col=c("red","yellow"),ylab="expression",las=2,
        main="TH1")
288 boxplot(newdat[,ord][,17:20],col=c("red","yellow"),ylab="expression",las=2,
        main="TH2")
289 boxplot(newdat[,ord][,21:26],col=c("red","yellow"),ylab="expression",las=2,
        main="TH17")
290 boxplot(newdat[,ord][,27:32],col=c("red","yellow"),ylab="expression",las=2,
        main="Immunoregulation")
291 #legend("bottomleft",c("blood","fluid"),fill=c("red","yellow"),cex=.8,horiz=
        T)
292
293 boxplot(newdat[,ord][,33:42],col=c("red","yellow"),ylab="expression",las=2,
        main="Neutrophil")
294 boxplot(newdat[,ord][,43:48],col=c("red","yellow"),ylab="expression",las=2,
        main="Growth factors")
295 boxplot(newdat[,ord][,49:62],col=c("red","yellow"),ylab="expression",las=2,
        main="Fibrosis")
296 boxplot(newdat[,ord][,63:84],col=c("red","yellow"),ylab="expression",las=2,
        main="MMPs")
297 #legend(15.5,-10,c("blood","fluid"),fill=c("red","yellow"))par(parbackup)
298 export.plot(file.prefix=paste("fig",figure,".","boxplot"),export.formats=
        export.formats.plots,height=height*3,width=(1 + sqrt(5))/2*height*1.5)
299
300 figure<-figure+1
301 par(parbackup)
302
303 genes<-colnames(datamat)
304 pvals.gene.by.compartment<-vector()
305 for (i in 1:42){
306   pvals.gene.by.compartment[i]<-wilcox.test(datamat[1:27,i],datamat[28:54,i
        ])$p.value
307 }
308 bonf<-p.adjust(pvals.gene.by.compartment,method="bonferroni")
309 bh<-p.adjust(pvals.gene.by.compartment,method="BH")
310
311 simple.comp.results<-data.frame(genes,pvals.gene.by.compartment,bonf,bh)
312 write.csv(simple.comp.results,file.path(dir.results,"genes_by_compartment.
        csv"))
313
314 # FIGURE 1 A:  Clustered raw data heatmaps (microarray style) ####
315 #heatmap.plus(t(datamat),scale="none",ColSideColors=phenomatrix,
        RowSideColors=genecol,col=redgreen(30),cexCol=1.2,cexRow=1,margins=c
        (10,30))
316 par(parbackup)
317 heatmap.plus(t(datamat),scale="none",ColSideColors=phenomatrix,col=redgreen
        (30),cexCol=1.2,cexRow=1,margins=c(10,24))
318 #HC using euclidean distance, complete linkage
319 s=round(seq(from=min(datamat),to=max(datamat),by=1),1)
320 colvec=redgreen(length(s))
321 legend("right",as.character(s), fill=colvec, ncol = 1, cex = .8,y.intersp
        =.8,title="LFC")
322
```

```
323
324 #experimental: calibrated heatmap
325 nr<-nrow(t(datamat))
326 hi<-rep(ref.max,nr)
327 lo<-rep(ref.min,nr)
328 caldata<-cbind(hi,lo,t(datamat))
329 heatmap.plus(caldata,scale="none",ColSideColors=calphenomatrix,col=redgreen
    (30),cexCol=1.2,cexRow=1,margins=c(10,24))
330 s=round(seq(from=min(datamat),to=max(datamat),by=1),1)
331 colvec=redgreen(length(s))
332 legend("right",as.character(s), fill=colvec, ncol = 1, cex = .8,y.intersp
    =.8,title="LFC")
333 export.plot(file.prefix=paste("fig",figure,".","all_genes_heatmap"),export.
    formats=export.formats.plots,height=height*1.5,width=(1 + sqrt(5))/2*
    height*1.5)
334 figure<-figure+1
335
336 par(parbackup)
337 # FIGURE 1 B and C: Correlation matrices####
338 #GENES
339 #the gene expression levels are both positively and negatively correlated
340 #cormat<-cor(datamat,method="pearson")
341 #heatmap_2(cormat,col=redgreen(300),scale="none",legend=2,legfrac=8,trim
    =.01)
342
343 cormatB<-cor(bloodmat,method="pearson")
344 #heatmap_2(cormatB,col=bluered(34),scale="none",legend=2,legfrac=8)
345 heatmap.2(cormatB,col=bluered(34),symkey=F,symbreaks=T,trace="none",density.
    info="none")
346 export.plot(file.prefix=paste("fig",figure,".","cormat_B"),export.formats=
    export.formats.plots,height=height*1.5,width=height*1.5)
347 figure<-figure+1
348
349 cormatF<-cor(fluidmat,method="pearson")
350 #heatmap_2(cormatF,col=bluered(34),scale="none",legend=2,legfrac=8)
351 heatmap.2(cormatF,col=bluered(34),symkey=F,symbreaks=T,trace="none",density.
    info="none")
352 export.plot(file.prefix=paste("fig",figure,".","cormat_F"),export.formats=
    export.formats.plots,height=height*1.5,width=height*1.5)
353 figure<-figure+1
354
355 library(plotrix)
356 cormatB2<-rescale(cormatB,c(-1,1))
357
358 #Data filtering step####
359
360 #non-specific filtering: all
361 ct=38
362 f1<-kOverA(floor(.95*nrow(datamat)),log(2^-ct,10))
363 flist<-filterfun(f1)
364 ans<-genefilter(t(datamat),flist)
365 datamat.filter<-t(t(datamat)[ans,])
366
367 f2<-kOverA(floor(.95*nrow(datamat[1:27,])),log(2^-ct,10))
368 flist.blood<-filterfun(f2)
369 ans.blood<-genefilter(t(datamat[1:27,]),flist.blood)
370 bloodmat.filter<-t(t(datamat[1:27,])[ans.blood,])
371
372 f3<-kOverA(floor(.95*nrow(datamat[28:54,])),log(2^-ct,10))
```

```
flist.fluid<-filterfun(f3)
ans.fluid<-genefilter(t(datamat[28:54,]),flist.fluid)
fluidmat.filter<-t(t(datamat[28:54,])[ans.fluid,])
#FIGURE S1A Clustered raw data heatmaps (microarray style) ##Filtered raw data####
par(parbackup)
#heatmap.plus(t(datamat),scale="none",ColSideColors=phenomatrix,RowSideColors=genecol,col=redgreen(30),cexCol=1.2,cexRow=1,margins=c(10,30))
heatmap.plus(t(datamat.filter),scale="none",ColSideColors=phenomatrix,col=redgreen(30),cexCol=1.2,cexRow=1,margins=c(10,24))
#HC using euclidean distance, complete linkage
s=round(seq(from=min(datamat.filter),to=max(datamat.filter),by=1),1)
colvec=redgreen(length(s))
legend("right",as.character(s), fill=colvec, ncol = 1, cex = .8,y.intersp=.8,title="LFC")


nr.filter<-nrow(t(datamat.filter))
hi.filter<-rep(ref.max,nr.filter)
lo.filter<-rep(ref.min,nr.filter)
caldata.filter<-cbind(hi.filter,lo.filter,t(datamat.filter))
heatmap.plus(caldata.filter,scale="none",ColSideColors=calphenomatrix,col=redgreen(30),cexCol=1.2,cexRow=1,margins=c(10,24))
s=round(seq(from=min(datamat),to=max(datamat),by=1),1)
colvec=redgreen(length(s))
legend("right",as.character(s), fill=colvec, ncol = 1, cex = .8,y.intersp=.8,title="LFC")
export.plot(file.prefix=paste("fig",figure,".","filtered_genes_heatmap"),export.formats=export.formats.plots,height=height*1.5,width=(1 + sqrt(5))/2*height*1.5)
figure<-figure+1

#FIGURE S1 B and C: Correlation matrices ##Filtered raw data####
#GENES
#the gene expression levels are both positively and negatively correlated

cormatB.filter<-cor(bloodmat.filter,method="pearson")
heatmap_2(cormatB.filter,col=bluered(34),scale="none",legend=2,legfrac=8)
export.plot(file.prefix=paste("fig",figure,".","filtered_genes_correlationMatrix_blood"),export.formats=export.formats.plots,height=height*1.5,width=(1 + sqrt(5))/2*height*1.5)
figure<-figure+1

cormatF.filter<-cor(fluidmat.filter,method="pearson")
heatmap_2(cormatF.filter,col=bluered(34),scale="none",legend=2,legfrac=8,trim=.01)
export.plot(file.prefix=paste("fig",figure,".","filtered_genes_correlationMatrix_fluid"),export.formats=export.formats.plots,height=height*1.5,width=(1 + sqrt(5))/2*height*1.5)
figure<-figure+1

#Moderated test statistics for blood versus fluid####

#Design and contrast matrices for all samples and subsets
comp<-as.factor(c(rep("Blood",27),rep("Fluid",27)))
design=model.matrix(~-1+factor(comp))
colnames(design)<-levels(comp)
design
```

```
contrast<-makeContrasts(Fluid-Blood,levels=design)
contrast

#Linear model: blood vs fluid
fit<-lmFit(t(datamat.filter),design)
fit2<-contrasts.fit(fit,contrast)
fit2<-eBayes(fit2)

par(parbackup)
gr<-decideTests(fit2,adjust.method="none")
vennDiagram(gr)
vc<-vennCounts(gr)

gr2<-decideTests(fit2,adjust.method="fdr")
vc2<-vennCounts(gr2)
vennDiagram(gr2)
#Results####
#DE
result1<-topTable(fit2,number=vc2[,2][2],adjust.method="BH",coef=1,sort.by="
    P",resort.by="logFC")
result1
#FIGURE 4: Volcano plot####
par(parbackup)
volcanoplot(fit2,coef=1,highlight=vc2[,2][2],cex=1,pch=20,col="red")
abline(v=0,col="blue",lty="dashed")
export.plot(file.prefix=paste("fig",figure,".","selected_genes_volcanoPlot")
    ,export.formats=export.formats.plots,height=height*1.5,width=(1 + sqrt
    (5))/2*height*1.5)
figure<-figure+1
#Table 2 and full result tables####

resultFullTable<-xtable(result1,digits=c(NA,NA,2,3,-3,-3,2))
write.csv(resultFullTable,file.path(dir.results,"SelectedGeneTopTableFull.
    csv"))

#Table 2
result1table<-xtable(result1[,c(1,2,4,5)],digits=c(NA,NA,2,-3,-3))
write.csv(result1table,file.path(dir.results,"SelectedGeneTopTable.csv"))
#print(result1table,floating=FALSE,include.rownames=FALSE)

#FIGURE 2A: Clustered result data heatmap (microarray style) results####
par(parbackup)
heatmap.plus(t(datamat.filter)[result1$ID,],scale="none",ColSideColors=
    phenomatrix,col=redgreen(30),cexCol=1.2,cexRow=1,margins=c(10,24))
#HC using euclidean distance, complete linkage
s=round(seq(from=min(datamat.filter[,result1$ID]),to=max(datamat.filter[,
    result1$ID]),by=1),1)
colvec=redgreen(length(s))
legend("right",as.character(s), fill=colvec, ncol = 1, cex = .8,y.intersp
    =.8,title="LFC")

nr.select<-nrow(t(datamat.filter)[result1$ID,])
hi.select<-rep(ref.max,nr.select)
lo.select<-rep(ref.min,nr.select)
caldata.select<-cbind(hi.select,lo.select,t(datamat.filter)[result1$ID,])
heatmap.plus(caldata.select,scale="none",ColSideColors=calphenomatrix,col=
    redgreen(30),cexCol=1.2,cexRow=1,margins=c(10,24))
s=round(seq(from=min(datamat),to=max(datamat),by=1),1)
colvec=redgreen(length(s))
```

```
469 legend("right",as.character(s), fill=colvec, ncol = 1, cex = .8,y.intersp
    =.8,title="LFC")
470 export.plot(file.prefix=paste("fig",figure,".","selected_genes_heatmap"),
    export.formats=export.formats.plots,height=height*1.5,width=(1 + sqrt(5)
    )/2*height*1.5)
471 figure<-figure+1
472
473 #FIGURE 2 B and C: Correlation matrices for selected genes####
474
475 cormatB.select<-cor(bloodmat.filter[,result1$ID],method="pearson")
476 heatmap_2(cormatB.select,col=bluered(34),scale="none",legend=2,legfrac=8,
    trim=.01)
477 export.plot(file.prefix=paste("fig",figure,".","selected_genes_
    correlationMatrix_blood"),export.formats=export.formats.plots,height=
    height*1.5,width=(1 + sqrt(5))/2*height*1.5)
478 figure<-figure+1
479
480 cormatF.select<-cor(fluidmat.filter[,result1$ID],method="pearson")
481 heatmap_2(cormatF.select,col=bluered(34),scale="none",legend=2,legfrac=8,
    trim=.01)
482 export.plot(file.prefix=paste("fig",figure,".","selected_genes_
    correlationMatrix_fluid"),export.formats=export.formats.plots,height=
    height*1.5,width=(1 + sqrt(5))/2*height*1.5)
483 figure<-figure+1
484
485 #FIGURE 3 PCA (oompa package)####
486 par(parbackup)
487 par(mfrow=c(1,3),mar=c(10,10,6,6))
488 par(xaxt="s")
489
490 #all genes
491 l=rownames(datamat)
492 trueClasses <- as.factor(comp)
493 spca <- SamplePCA(t(datamat), trueClasses)
494 plot(spca, col = c("red", "blue"),main="All genes",cex=2,cex.axis=2,cex.main
    =3,cex.lab=2.5)
495 mtext(sprintf("%.2f",spca@variances[1]/sum(spca@variances)),side=1,line=3,
    adj=1,cex=1.5)
496 mtext(sprintf("%.2f",spca@variances[2]/sum(spca@variances)),side=2,line=3,
    adj=1,cex=1.5)
497 # mark the group centers
498 x1 <- predict(spca, matrix(apply(t(datamat[grep("b",rownames(datamat)),]),
    1, mean), ncol=1))
499 points(x1[1], x1[2], col='red', cex=6,pch=9)
500 x2 <- predict(spca, matrix(apply(t(datamat[grep("f",rownames(datamat)),]),
    1, mean), ncol=1))
501 points(x2[1], x2[2], col='blue', cex=6,pch=9)
502
503 #filtered genes
504 l=rownames(datamat.filter)
505 trueClasses1 <- as.factor(comp)
506 spca1 <- SamplePCA(t(datamat.filter), trueClasses1)
507 plot(spca1, col = c("red", "blue"),main="Filtered genes",cex=2,cex.axis=2,
    cex.main=3,cex.lab=2.5)
508 mtext(sprintf("%.2f",spca1@variances[1]/sum(spca1@variances)),side=1,line=3,
    adj=1,cex=1.5)
509 mtext(sprintf("%.2f",spca1@variances[2]/sum(spca1@variances)),side=2,line=3,
    adj=1,cex=1.5)
510 # mark the group centers
```

```
511 x1 <- predict(spca1, matrix(apply(t(datamat.filter[grep("b",rownames(datamat
    ))),]), 1, mean), ncol=1))
512 points(x1[1], x1[2], col='red', cex=6,pch=9)
513 x2 <- predict(spca1, matrix(apply(t(datamat.filter[grep("f",rownames(datamat
    ))),]), 1, mean), ncol=1))
514 points(x2[1], x2[2], col='blue', cex=6,pch=9)
515
516 #selected genes
517 trueClasses2 <- as.factor(comp)
518 spca2 <- SamplePCA(t(datamat.filter)[result1$ID,], trueClasses2)
519 plot(spca2, col = c("red", "blue"),main="Selected genes",cex=2,cex.axis=2,
    cex.main=3,cex.lab=2.5)
520 mtext(sprintf("%.2f",spca2@variances[1]/sum(spca2@variances)),side=1,line=3,
    adj=1,cex=1.5)
521 mtext(sprintf("%.2f",spca2@variances[2]/sum(spca2@variances)),side=2,line=3,
    adj=1,cex=1.5)
522 # mark the group centers
523 x1 <- predict(spca2, matrix(apply(t(datamat.filter[grep("b",rownames(datamat
    ))),])[result1$ID,], 1, mean), ncol=1))
524 points(x1[1], x1[2], col='red', cex=6,pch=9)
525 x2 <- predict(spca2, matrix(apply(t(datamat.filter[grep("f",rownames(datamat
    ))),])[result1$ID,], 1, mean), ncol=1))
526 points(x2[1], x2[2], col='blue', cex=6,pch=9)
527
528
529 sprintf("%.10f",0.25)
530
531 export.plot(file.prefix=paste("fig",figure,".","PCA of 3 data subsets"),
    export.formats=export.formats.plots,height=height*1.5,width=2*(1 + sqrt
    (5))/2*height*1.5)
532 figure<-figure+1
533
534 par(parbackup)
535 #screeplot(spca)
536 #screeplot(spca1)
537 #screeplot(spca1,type="lines")
538 #screeplot(spca2)
539
540 #TABLE S1: Phenodata
541
542 phenodataTS2<-phenodata[,c(2,4,5,12,13,15,14,26,27,28,35,36,37,38)]
543 write.csv(phenodataTS2,file.path(dir.results,"TABLES2.csv"))
544 #pdat<-xtable(phenodataTS2)
545 #print(pdat)
546
547
548 #Protein
549
550 subjects<-rownames(datamat)
551 #Blood only
552 data3blood.prot<-data3.prot[data3.prot$compart=="blood",]
553 #Fluid only
554 data3fluid.prot<-data3.prot[data3.prot$compart=="fluid",]
555
556 bloodmat.p<-matrix(as.numeric(as.matrix(data3blood.prot[,-40:-1])),nrow=25)
557 rownames(bloodmat.p)<-data3blood.prot$SampleID
558 fluidmat.p<-matrix(as.numeric(as.matrix(data3fluid.prot[,-40:-1])),nrow=25)
559 rownames(fluidmat.p)<-data3fluid.prot$SampleID
560
```

```
561 datamat.p<-as.matrix(rbind(bloodmat.p,fluidmat.p))
562 colnames(datamat.p)<-c("IL1B","IL6","IL18","TNF","IFNG","CXCL10","IL13","
    IL17","IL22","IL23A","TGFB1","IL10","IL8","IL12P70","GAL3","ACSDKP","
    TIMP1","TIMP2","MMP2TIMP1","MMP9TIMP2","MMP1","MMP2","MMP3","MP7","MMP8"
    ,"MMP9")
563
564 datamat.p.sel<-datamat.p[,is.element(colnames(datamat.p),result1$ID)]
565 ord.p<-interleave(seq(1,9),seq(10,18))
566 newdat.p<-cbind(datamat.p.sel[1:25,],datamat.p.sel[26:50,])
567 par(parbackup)
568 boxplot(newdat.p[,ord.p],col=c("red","yellow"),las=2,main="Levels of 9
    proteins")
569 legend("topright",c("blood","fluid"),fill=c("red","yellow"),cex=.8)
570
571 datamat.p.notsel<-datamat.p[,!is.element(colnames(datamat.p),result1$ID)]
572 ord.p.notsel<-interleave(seq(1,17),seq(18,34))
573 newdat.p.notsel<-cbind(datamat.p.notsel[1:25,],datamat.p.notsel[26:50,])
574 par(parbackup)
575 boxplot(newdat.p.notsel[,ord.p.notsel],col=c("red","yellow"),las=2,main="
    Levels of other proteins")
576 legend("topright",c("blood","fluid"),fill=c("red","yellow"),cex=.8)
577
578 #plot both on one page (log10 scale)
579 par(mfrow=c(2,1))
580 boxplot(log10(newdat.p[,ord.p]+1),col=c("red","yellow"),las=2,main="Levels
    of 9 proteins (where mRNA was DE)")
581 legend("topleft",c("bl","fl"),fill=c("red","yellow"),cex=.7)
582 boxplot(log10(newdat.p.notsel[,ord.p.notsel]+1),col=c("red","yellow"),las=2,
    main="Levels of other proteins (where mRNA was not DE)")
583 legend("topleft",c("bl","fl"),fill=c("red","yellow"),cex=.7)
584 par(parbackup)
585 export.plot(file.prefix=paste("fig",figure,".","boxplots of proteins by
    compartment"),export.formats=export.formats.plots,height=height*1.5,
    width=(1 + sqrt(5))/2*height*1.5)
586 figure<-figure+1
587
588 proteins<-colnames(datamat.p)
589 pvals.protein.by.compartment<-vector()
590 for (i in 1:26){
591   try(pvals.protein.by.compartment[i]<-wilcox.test(datamat.p[1:25,i],datamat
    .p[26:50,i])$p.value)
592 }
593 bonf.prot<-p.adjust(pvals.protein.by.compartment,method="bonferroni")
594 bh.prot<-p.adjust(pvals.protein.by.compartment,method="BH")
595 simple.comp.results.prot<-data.frame(proteins,pvals.protein.by.compartment,
    bonf.prot,bh.prot)
596 write.csv(simple.comp.results.prot,file.path(dir.results,"proteins_by_
    compartment.csv"))
597
598 proteins.sel<-colnames(datamat.p.sel)
599 pvals.protein.by.compartment.sel<-vector()
600 for (i in 1:ncol(datamat.p.sel)){
601   try(pvals.protein.by.compartment.sel[i]<-wilcox.test(datamat.p.sel[1:25,i
    ],datamat.p.sel[26:50,i])$p.value)
602 }
603 bonf.prot.sel<-p.adjust(pvals.protein.by.compartment.sel,method="bonferroni"
    )
604 bh.prot.sel<-p.adjust(pvals.protein.by.compartment.sel,method="BH")
```

```
simple.comp.results.prot.sel<-data.frame(proteins.sel,pvals.protein.by.compartment.sel,bonf.prot.sel,bh.prot.sel)
write.csv(simple.comp.results.prot.sel,file.path(dir.results,"selected_proteins_by_compartment.csv"))

proteins.notsel<-colnames(datamat.p.notsel)
pvals.protein.by.compartment.notsel<-vector()
for (i in 1:ncol(datamat.p.notsel)){
  try(pvals.protein.by.compartment.notsel[i]<-wilcox.test(datamat.p.notsel[1:25,i],datamat.p.notsel[26:50,i])$p.value)
}
bonf.prot.notsel<-p.adjust(pvals.protein.by.compartment.notsel,method="bonferroni")
bh.prot.notsel<-p.adjust(pvals.protein.by.compartment.notsel,method="BH")
simple.comp.results.prot.notsel<-data.frame(proteins.notsel,pvals.protein.by.compartment.notsel,bonf.prot.notsel,bh.prot.notsel)
write.csv(simple.comp.results.prot.notsel,file.path(dir.results,"non_selected_proteins_by_compartment.csv"))

#median ratios of proteins
apply(datamat.p.sel[26:50,],2,median,na.rm=T)/apply(datamat.p.sel[1:25,],2,median,na.rm=T)

#THE END OF TABLES AND FIGURES FOR THE PAPER####
#NOTHING BELOW IS SHOWN IN PAPER BUT LACK OF HIV EFFECT NEEDS TO BE MENTIONED!

#Moderated test statistics for blood ONLY####

#Contrast for blood
#HIV status
comp<-as.factor(data3blood$HIV)
levels(comp)<-c("Neg","Pos")
design=model.matrix(~-1+factor(comp))
colnames(design)<-levels(comp)
design
contrast<-makeContrasts(Pos-Neg,levels=design)
contrast

#Linear model: blood
fit<-lmFit(t(bloodmat.filter),design)
fit2<-contrasts.fit(fit,contrast)
fit2<-eBayes(fit2)

gr<-decideTests(fit2,adjust.method="none")
vennDiagram(gr)
vc<-vennCounts(gr)
gr2<-decideTests(fit2,adjust.method="fdr")
vc2<-vennCounts(gr2)
vennDiagram(gr2)

#this shows that 2 genes are DE after BH correction

#Results
#DE without correction
result1<-topTable(fit2,number=vc[,2][2],adjust="fdr",coef=1,sort.by="P")
result1

#Clustered result data heatmap (microarray style) blood only####
```

```
heatmap.plus(t(bloodmat.filter)[result1$ID,],scale="none",ColSideColors=
    phenomatrix.blood,col=redgreen(24))

#Moderated test statistics for fluid ONLY####

#Contrasts and linear models for fluid

#HIV status
comp<-as.factor(data3fluid$HIV)
levels(comp)<-c("Neg","Pos")
design=model.matrix(~-1+factor(comp))
colnames(design)<-levels(comp)
design
contrast<-makeContrasts(Pos-Neg,levels=design)
contrast

fit<-lmFit(t(fluidmat.filter),design)
fit2<-contrasts.fit(fit,contrast)
fit2<-eBayes(fit2)

gr<-decideTests(fit2,adjust.method="none")
vennDiagram(gr)
vc<-vennCounts(gr)
gr2<-decideTests(fit2,adjust.method="fdr")
vc2<-vennCounts(gr2)
vennDiagram(gr2)

#Low CD4
comp<-as.factor(data3fluid$CD4<200)
levels(comp)<-c("high","low")
design=model.matrix(~-1+factor(comp))
colnames(design)<-levels(comp)
design
contrast<-makeContrasts(high-low,levels=design)
contrast

fit<-lmFit(t(fluidmat.filter),design)
fit2<-contrasts.fit(fit,contrast)
fit2<-eBayes(fit2)

gr<-decideTests(fit2,adjust.method="none")
vennDiagram(gr)
vc<-vennCounts(gr)
gr2<-decideTests(fit2,adjust.method="fdr")
vc2<-vennCounts(gr2)
vennDiagram(gr2)

#Results: Nothing is DE by HIV status or low CD4. The genes are not HIV
    genes...
```

Listing A.8: RT-PCR code

A.5 Relation of IMPI-MA data to BERRY dataset

```
##TO DO
#1. table 1
#2. export probe lists (not annotated)
#3. annotate probe lists and export

#setup####
source("/Users/armindeffur/Documents/002_Science_universal/Active_Research/IMPI-Microarray/Projects/01_scripts/06_IMPI-MA/Analysis/00_System/System_setup.R")
source(file.path(dir.R.files,"superHeatmap.R"))
par("xpd"=FALSE)
## Individual data folders (specific for each part)
dir.rdata <- file.path(dir.data_root,'09_IMPI_MA_DATA/05_RData')

#Output folders
dir.home<-dir.root
dir.main <- file.path(dir.root, '02_output/06_IMPI-MA/DifferentialExpression','DE_Baylor')

#Current date string used for versioning output
ds<-Sys.time()

#versioned output
dir.output.version<-file.path(dir.main,paste("version_",ds))

## Define folder for storing results NB do this for each analysis
dir.results <- file.path(dir.output.version, "results")
## Define folder for saving figures
dir.figures <- file.path(dir.output.version, "figures")
for (dir in c(dir.main, dir.figures, dir.results)) {
  if (!file.exists(dir)) {
    dir.create(dir, recursive=T,showWarnings=T)
  }
}


#sink(type="output",file=file.path(dir.results,"session_output.txt"))
sessionInfo()
print(paste("Begin time:",ds))
print(paste("Results will be saved to", dir.results))


print(paste("Figures will be saved to", dir.figures))



#load data####
load(file.path(dir.rdata,'berry_train.RData'))
load(file.path(dir.rdata,'berry_val.RData'))
load(file.path(dir.rdata,'blood_hivNeg_aTB_noTB.RData'))


#Average Normalization
#This replicates the data output by GenomeStudio, starting with raw, non-normalized summary data

```

```
datalist<-list("SA blood IMPI hiv neg"=B.N.AX,"SA blood VAL hiv neg"=berry.val,"Berry Training Set"=berry.train)
#scaling normalization (as in SOFT data)
for (i in 1:length(datalist)){
datalist[[i]]<-ssn(datalist[[i]],scaling=TRUE,bgMethod='none',fgMethod='mean')
}

########################################.

#Make Table 1####
########################################.

#Table 1: Compare groups across the 3 sets####

#####ALL
agedata<-list(as.numeric(pData(berry.train)$age),
              as.numeric(pData(berry.val)$age),
              as.numeric(pData(B.N.AX)$AGE_CALC))

lapply(agedata,length)

lapply(agedata,summary)
kruskal.test(agedata)

sexdata<-matrix(
  rbind(
  table(pData(berry.train)$sex),
  table(pData(berry.val)$sex),
  table(pData(B.N.AX)$SEX)
  ),ncol=2)
rownames(sexdata)<-c("Berry.train","Berry.val","B.N.AX");colnames(sexdata)<-c("Female","Male")
prop.test(sexdata)

table(pData(berry.train)$CrudeEthnicity)
table(pData(berry.val)$CrudeEthnicity)
table(pData(B.N.AX)$ETHNICITY)

ethnicity<-matrix(
  rbind(
    c(11,17,7,6,0,0,1),
    c(51,0,0,0,0,0,0),
    c(21,0,0,0,0,1,0)
    ),ncol=7)
rownames(ethnicity)<-c("Berry.train","Berry.val","B.N.AX");colnames(ethnicity)<-c("Black","White","Asian other","South Asian","Coloured","Indian","Other")
fisher.test(ethnicity,simulate.p.value=TRUE)

table(pData(berry.train)$threeclass)
table(pData(berry.val)$threeclass)
table(pData(B.N.AX)$class)

twoclass<-matrix(
  rbind(
    c(13,29),
    c(20,31),
    c(12,10)
```

```
        ),ncol=2)
colnames(twoclass)<-c("Active TB","Not active TB")
rownames(twoclass)<-c("Berry.train","Berry.val","B.N.AX")
prop.test(twoclass)

fourclass<-matrix(
  rbind(
    c(12,17,13,0),
    c(0,31,20,0),
    c(5,5,4,8)
  ),ncol=4)
colnames(fourclass)<-c("Healthy","LTBI","PTB","TB-PC")
rownames(fourclass)<-c("Berry.train","Berry.val","B.N.AX")
fisher.test(fourclass,simulate.p.value=TRUE)

#####healthy
agedata.h<-list(as.numeric(pData(berry.train)[pData(berry.train)$threeclass
    =="Control",]$age),
                #as.numeric(pData(berry.val)[pData(berry.val)$threeclass=="
    Control",]$age),
                as.numeric(pData(B.N.AX)[pData(B.N.AX)$class=="CON_healthy",]$
    AGE_CALC))

lapply(agedata.h,length)

lapply(agedata.h,summary)
kruskal.test(agedata.h)

tstdata.h<-list(as.numeric(pData(berry.train)[pData(berry.train)$threeclass
    =="Control",]$TST),
                  #as.numeric(pData(berry.val)[pData(berry.val)$threeclass=="
    Control",]$age),
                  as.numeric(pData(B.N.AX)[pData(B.N.AX)$class=="CON_healthy"
    ,]$TST))

lapply(tstdata.h,length)

lapply(tstdata.h,summary)
kruskal.test(tstdata.h)

sexdata.h<-matrix(
  rbind(
    table(pData(berry.train)[pData(berry.train)$threeclass=="Control",]$sex)
    ,
    #table(pData(berry.val)$sex),
    table(pData(B.N.AX)[pData(B.N.AX)$class=="CON_healthy",]$SEX)
  ),ncol=2)
rownames(sexdata.h)<-c("Berry.train","B.N.AX");colnames(sexdata.h)<-c("
    Female","Male")
prop.test(sexdata.h)

table(pData(berry.train)[pData(berry.train)$threeclass=="Control",]$
    CrudeEthnicity)
#table(pData(berry.val)$CrudeEthnicity)
table(pData(B.N.AX)[pData(B.N.AX)$class=="CON_healthy",]$ETHNICITY)

ethnicity.h<-matrix(
  rbind(
    c(0,12,0,0,0,0,0),
```

```
157     #c(51,0,0,0,0,0,0),
158     c(5,0,0,0,0,0,0)
159   ),ncol=7)
160 rownames(ethnicity.h)<-c("Berry.train","B.N.AX");colnames(ethnicity.h)<-c("
      Black","White","Asian other","South Asian","Coloured","Indian","Other")
161 fisher.test(ethnicity.h,simulate.p.value=TRUE)
162
163
164 #####LTBI
165 agedata.l<-list(as.numeric(pData(berry.train)[pData(berry.train)$threeclass
      =="Latent",]$age),
166                 as.numeric(pData(berry.val)[pData(berry.val)$threeclass=="
      Latent",]$age),
167                 as.numeric(pData(B.N.AX)[pData(B.N.AX)$class=="CON_LTBI",]$
      AGE_CALC))
168
169 lapply(agedata.l,length)
170
171 lapply(agedata.l,summary)
172 kruskal.test(agedata.l)
173
174 tstdata.l<-list(as.numeric(pData(berry.train)[pData(berry.train)$threeclass
      =="Latent",]$TST),
175                 as.numeric(pData(berry.val)[pData(berry.val)$threeclass=="
      Latent",]$TST),
176                 as.numeric(pData(B.N.AX)[pData(B.N.AX)$class=="CON_LTBI",]$
      TST))
177
178 lapply(tstdata.l,length)
179
180 lapply(tstdata.l,summary)
181 kruskal.test(tstdata.l)
182
183 sexdata.l<-matrix(
184   rbind(
185     table(pData(berry.train)[pData(berry.train)$threeclass=="Latent",]$sex),
186     table(pData(berry.val)[pData(berry.val)$threeclass=="Latent",]$sex),
187     table(pData(B.N.AX)[pData(B.N.AX)$class=="CON_LTBI",]$SEX)
188   ),ncol=2)
189 rownames(sexdata.l)<-c("Berry.train","Berry.val","B.N.AX");colnames(sexdata.
      l)<-c("Female","Male")
190 prop.test(sexdata.l)
191
192 table(pData(berry.train)[pData(berry.train)$threeclass=="Latent",]$
      CrudeEthnicity)
193 table(pData(berry.val)[pData(berry.val)$threeclass=="Latent",]$
      CrudeEthnicity)
194 table(pData(B.N.AX)[pData(B.N.AX)$class=="CON_LTBI",]$ETHNICITY)
195
196 ethnicity.l<-matrix(
197   rbind(
198     c(7,2,5,3,0,0,0),
199     c(31,0,0,0,0,0,0),
200     c(5,0,0,0,0,0,0)
201   ),ncol=7)
202 rownames(ethnicity.l)<-c("Berry.train","Berry.val","B.N.AX");colnames(
      ethnicity.l)<-c("Black","White","Asian other","South Asian","Coloured","
      Indian","Other")
203 fisher.test(ethnicity.l,simulate.p.value=TRUE)
```

```
204
205 #####TB
206 agedata.t<-list(as.numeric(pData(berry.train)[pData(berry.train)$threeclass
    =="PTB",]$age),
207                 as.numeric(pData(berry.val)[pData(berry.val)$threeclass=="
    PTB",]$age),
208                 as.numeric(pData(B.N.AX)[pData(B.N.AX)$class3=="activeTB",]$
    AGE_CALC))
209
210 lapply(agedata.t,length)
211
212 lapply(agedata.t,summary)
213 kruskal.test(agedata.t)
214
215 sexdata.t<-matrix(
216   rbind(
217     table(pData(berry.train)[pData(berry.train)$threeclass=="PTB",]$sex),
218     table(pData(berry.val)[pData(berry.val)$threeclass=="PTB",]$sex),
219     table(pData(B.N.AX)[pData(B.N.AX)$class3=="activeTB",]$SEX)
220   ),ncol=2)
221 rownames(sexdata.t)<-c("Berry.train","Berry.val","B.N.AX");colnames(sexdata.
    t)<-c("Female","Male")
222 prop.test(sexdata.t)
223
224 table(pData(berry.train)[pData(berry.train)$threeclass=="PTB",]$
    CrudeEthnicity)
225 table(pData(berry.val)[pData(berry.val)$threeclass=="PTB",]$CrudeEthnicity)
226 table(pData(B.N.AX)[pData(B.N.AX)$class3=="activeTB",]$ETHNICITY)
227
228 ethnicity.t<-matrix(
229   rbind(
230     c(4,3,2,3,0,0,1),
231     c(20,0,0,0,0,0,0),
232     c(11,0,0,0,0,1,0)
233   ),ncol=7)
234 rownames(ethnicity.t)<-c("Berry.train","Berry.val","B.N.AX");colnames(
    ethnicity.t)<-c("Black","White","Asian other","South Asian","Coloured","
    Indian","Other")
235 fisher.test(ethnicity.l,simulate.p.value=TRUE)
236
237 classdata.t<-matrix(
238   rbind(
239     table(factor(pData(berry.train)[pData(berry.train)$threeclass=="PTB",]$
    threeclass,levels=c("PTB","TB-PC"))),
240     table(factor(pData(berry.val)[pData(berry.val)$threeclass=="PTB",]$
    threeclass,levels=c("PTB","TB-PC"))),
241     table(pData(B.N.AX)[pData(B.N.AX)$class3=="activeTB",]$class)
242   ),ncol=2)
243 rownames(classdata.t)<-c("Berry.train","Berry.val","B.N.AX");colnames(
    classdata.t)<-c("PTB","TB-PC")
244 prop.test(classdata.t)
245
246 tech<-matrix(
247   rbind(
248   table(factor(rep("Tempus",42),levels=c("Tempus","paxGene"))),
249   table(factor(rep("Tempus",51),levels=c("Tempus","paxGene"))),
250   table(factor(rep("paxGene",22),levels=c("Tempus","paxGene")))
251   ),ncol=2)
```

```
rownames(tech)<-c("Berry.train","Berry.val","B.N.AX");colnames(tech)<-c("Tempus","paxGene")
prop.test(tech)

#Table 1: Compare groups within each set####

#age
#train
age.train<-list(agedata.h[[1]],agedata.l[[1]],agedata.t[[1]])
lapply(age.train,length)
lapply(age.train,summary)
kruskal.test(age.train)

#val
age.val<-list(agedata.l[[2]],agedata.t[[2]])
lapply(age.val,length)
lapply(age.val,summary)
kruskal.test(age.val)

#bnax
age.bnax<-list(agedata.h[[2]],agedata.l[[3]],agedata.t[[3]])
lapply(age.bnax,length)
lapply(age.bnax,summary)
kruskal.test(age.bnax)

#sex
#train
sexdata.train<-matrix(
  rbind(
    table(pData(berry.train)[pData(berry.train)$threeclass=="Control",]$sex),
    table(pData(berry.train)[pData(berry.train)$threeclass=="Latent",]$sex),
    table(pData(berry.train)[pData(berry.train)$threeclass=="PTB",]$sex)
  ),ncol=2)
rownames(sexdata.train)<-c("Healthy","Latent","PTB");colnames(sexdata.train)<-c("Female","Male")
prop.test(sexdata.train)


#val
sexdata.val<-matrix(
  rbind(
    table(pData(berry.val)[pData(berry.val)$threeclass=="Latent",]$sex),
    table(pData(berry.val)[pData(berry.val)$threeclass=="PTB",]$sex)
    )
  ,ncol=2)
rownames(sexdata.val)<-c("Latent","PTB");colnames(sexdata.val)<-c("Female","Male")
prop.test(sexdata.val)

#bnax
sexdata.bnax<-matrix(
  rbind(
    table(pData(B.N.AX)[pData(B.N.AX)$class=="CON_healthy",]$SEX),
    table(pData(B.N.AX)[pData(B.N.AX)$class=="CON_LTBI",]$SEX),
    table(pData(B.N.AX)[pData(B.N.AX)$class3=="activeTB",]$SEX)
  ),ncol=2)
rownames(sexdata.bnax)<-c("Healthy","Latent","TB");colnames(sexdata.bnax)<-c("Female","Male")
```

```
prop.test(sexdata.bnax)

#ethnicity

ethnicity.train<-matrix(
  rbind(
    c(0,12,0,0,0,0,0),
    c(7,2,5,3,0,0,0),
    c(4,3,2,3,0,0,1)
  ),ncol=7)
rownames(ethnicity.train)<-c("Healthy","Latent","TB");colnames(ethnicity.
    train)<-c("Black","White","Asian other","South Asian","Coloured","Indian
    ","Other")
fisher.test(ethnicity.train,simulate.p.value=TRUE)

ethnicity.val<-matrix(
  rbind(
    #c(0,12,0,0,0,0,0),
    c(31,0,0,0,0,0,0),
    c(20,0,0,0,0,0,0)
  ),ncol=7)
rownames(ethnicity.val)<-c("Latent","TB");colnames(ethnicity.val)<-c("Black"
    ,"White","Asian other","South Asian","Coloured","Indian","Other")
#prop.test(ethnicity.val)#fails

ethnicity.bnax<-matrix(
  rbind(
    c(5,0,0,0,0,0,0),
    c(5,0,0,0,0,0,0),
    c(11,0,0,0,0,1,0)
  ),ncol=7)
rownames(ethnicity.bnax)<-c("Healthy","Latent","TB");colnames(ethnicity.bnax
    )<-c("Black","White","Asian other","South Asian","Coloured","Indian","
    Other")
fisher.test(ethnicity.bnax,simulate.p.value=FALSE)

tabledata<-read.csv("/Users/armindeffur/Documents/002_Science_universal/
    Active_Research/IMPI-Microarray/Documents/PhD/10_Thesis/Thesis_all/
    Tables/DE_1_table1.csv",sep=",")

#show data####

#boxplots and stripcharts
figure<-1
setwd(dir.figures)
#Need to figure out how to do this automatically. this fails if the order of
     samples in the input data is changed
tbstat=list(as.factor(B.N.AX$class2),as.factor(berry.val$confirmed_diagnosis
    ),as.factor(berry.train$confirmed_diagnosis))
levels(tbstat[[1]])<-c("green","deepskyblue","orange","red","red")
levels(tbstat[[2]])<-c("deepskyblue","orange")
levels(tbstat[[3]])<-c("green","green","deepskyblue","orange")
colnames(exprs(datalist[[1]]))<-B.N.AX$sample_name
colnames(exprs(datalist[[2]]))<-berry.val$sample_name
colnames(exprs(datalist[[3]]))<-berry.train$sample_name
# the order of this depends completely on the order in the csv file - it
    should not impact on the actual analysis. Just need to make sure that
    the class assignments for DE are correct
plotnum<-1
```

```
for (i in 1:length(datalist)){
boxplot(datalist[[i]],main=names(datalist)[i],col=as.character(tbstat[[i]]))
export.plot(file.prefix=paste('fig',figure,'.',plotnum,names(datalist)[i],'
    boxplot'),export.formats=export.formats.plots,height=height,width=(1 +
    sqrt(5))/2*height);plotnum<-plotnum+1
}

figure<-figure+1

#Outlier plot
plotnum<-1
for (i in 1:length(datalist)){
  plot(datalist[[i]],what="outlier",main=names(datalist)[i])
  export.plot(file.prefix=paste('fig',figure,'.',plotnum,names(datalist)[i],
    'outlier_plot'),export.formats=export.formats.plots,height=height,width
    =(1 + sqrt(5))/2*height);plotnum<-plotnum+1
}
figure<-figure+1

#MDS all genes
plotnum<-1
for (i in 1:length(datalist)){
plotSampleRelationsAD(datalist[[i]],method="mds",color=as.character(tbstat[[
    i]]),plotchar=16)
legend("bottomleft",
       list(c("Healthy","Latent","PTB","TB-PC"),c("Latent","PTB"),c("Healthy
    ","Latent","PTB"))[[i]],
       pch=16,
       col=list(c("green","deepskyblue","orange","red"),c("deepskyblue","
    orange"),c("green","deepskyblue","orange"))[[i]],
       cex=.8
)
export.plot(file.prefix=paste('fig',figure,".",plotnum,'mds_all',names(
    datalist[[i]])),export.formats=export.formats.plots,height=height*0.7,
    width=(1 + sqrt(5))/2*height*0.7);plotnum<-plotnum+1
}
figure<-figure+1

#PCA all genes
plotnum<-1
for (i in 1:length(datalist)){
l=colnames(exprs(datalist[[i]]))
trueClasses <-list(as.factor(B.N.AX$class2),as.factor(berry.val$confirmed_
    diagnosis),as.factor(berry.train$confirmed_diagnosis))
spca <- SamplePCA(exprs(datalist[[i]]), trueClasses[[i]])
plot(spca, col=list(c("deepskyblue","orange","green","red","red"),c("
    deepskyblue","orange"),c("green","green","deepskyblue","orange"))[[i]],
    main="All genes")#,cex=2,cex.axis=2,cex.main=3,cex.lab=2.5,pch=16)
mtext(sprintf("%.2f",spca@variances[1]/sum(spca@variances)),side=1,line=3,
    adj=1,cex=1.5)
mtext(sprintf("%.2f",spca@variances[2]/sum(spca@variances)),side=2,line=3,
    adj=1,cex=1.5)
legend("bottomleft",
       list(c("Healthy","Latent","PTB","TB-PC"),c("Latent","PTB"),c("Healthy
    ","Latent","PTB"))[[i]],
       pch=16,
       col=list(c("green","deepskyblue","orange","red"),c("deepskyblue","
    orange"),c("green","deepskyblue","orange"))[[i]],
       cex=.8
```

```
)
# mark the group centers
#x1 <- predict(spca, matrix(apply(t(exprs(datalist[[i]])[grep("b",rownames(
    exprs(datalist[[i]]))),]), 1, mean), ncol=1))
#points(x1[1], x1[2], col='red', cex=6,pch=9)
#x2 <- predict(spca, matrix(apply(t(exprs(datalist[[i]])[grep("f",rownames(
    exprs(datalist[[i]]))),]), 1, mean), ncol=1))
#points(x2[1], x2[2], col='blue', cex=6,pch=9)

export.plot(file.prefix=paste('fig',figure,'.',plotnum,'pca_all',names(
    datalist[[i]])),export.formats=export.formats.plots,height=height*0.7,
    width=(1 + sqrt(5))/2*height*0.7);plotnum<-plotnum+1
}
figure<-figure+1
##

########################################.
#Baylor method
########################################.

#preprocess####
#threshold then median scale raw data
for (i in 1:length(datalist)){
lower.thresh <- 10
data.raw <- exprs(datalist[[i]])
data.raw.thresh<-data.raw
data.raw.thresh[data.raw<lower.thresh]<-lower.thresh
genemedians <- apply(data.raw.thresh, 1, median)
data.ms <- data.raw.thresh / genemedians
exprs(datalist[[i]])<-data.ms
}

#boxplots of median scaled data
plotnum<-1
for (i in 1:length(datalist)){
  boxplot(datalist[[i]],main=names(datalist)[i],col=as.character(tbstat[[i
    ]]))
  export.plot(file.prefix=paste('fig',figure,'.',plotnum,names(datalist)[i],
    'boxplot.ms'),export.formats=export.formats.plots,height=height,width=(1
     + sqrt(5))/2*height);plotnum<-plotnum+1
}
figure<-figure+1

#profile plots of median scaled data (not implemented)
#plotnum<-1
#for (i in 1:length(datalist)){
#par(mar=c(10,7,2,1))
#matplot(log2(t(exprs(datalist[[i]]))),type="l",ylim=c(-7,7),ylab="
    Expression (log2)",xaxt='n',main=paste(names(datalist)[i],'profile plot
    all probes'));
#abline(a=1,b=0,col="yellow");
#abline(a=-1,b=0,col="yellow");
#abline(a=-0,b=0,col="yellow");
#axis(1,at=1:dim(exprs(datalist[[i]]))[[2]],lab=colnames(exprs(datalist[[i
    ]])),las=2)
#par(parbackup)
#export.plot(file.prefix=paste('fig',figure,'.',plotnum,names(datalist)[i],'
    profile_plot all probes'),export.formats=export.formats.plots,height=
    height,width=(1 + sqrt(5))/2*height);plotnum<-plotnum+1
```

```
#}
#figure<-figure+1

#filter####

#detection filter
detection.threshold <- 0.01
samples.threshold.fraction <- 0.1
present.probes<-list()
plotnum<-1
for (i in 1:length(datalist)){
data.detection<-detection(datalist[[i]])
samples.threshold <- samples.threshold.fraction * ncol(data.detection)
is.present <- data.detection < detection.threshold
# is.present is Boolean matrix with present/absent status for individual
    gene and chip
present.samples.per.probe <- apply(is.present, 1, sum)
# present.samples.per.probe is vector with number of chips that have present
    call for
# each probe
present.probes[[i]] <- present.samples.per.probe > samples.threshold
# present.probes is Boolean vector with present/absent status for each probe
    taken over all
# chips
print(paste(sum(present.probes[[i]]), "probes present in at least 10% of the
    samples"))

sample.breaks <- seq(-0.5, ncol(exprs(datalist[[i]]))+0.5, by=1)
h <- hist(present.samples.per.probe, breaks=sample.breaks,
          main=paste('Number of samples with "present" call per probe\n',
    names(datalist)[[i]]),
          xlab="Number of samples with present call", ylab="Number of probes
    ",
          col="#BBFFBB",labels=TRUE)
abline(v=floor(samples.threshold)+0.5, col='#880000', lwd=2)
export.plot(file.prefix=paste('fig',figure,'.',plotnum,names(datalist)[i],'
    present_call_histogram'),export.formats=export.formats.plots,height=
    height,width=(1 + sqrt(5))/2*height);plotnum<-plotnum+1

}

figure<-figure+1

#fold-change filter
samples.threshold.fraction <- 0.1
fold.threshold <-2
updown.probes<-list()
updata<-list()
downdata<-list()
selected.probes<-list()
plotnum<-1
for (i in 1:length(datalist)){

samples.threshold <- samples.threshold.fraction * ncol(exprs(datalist[[i]]))
    ;
is.up <- exprs(datalist[[i]]) > fold.threshold;

print(paste(sum(is.up), "up-regulated probes in",names(datalist)[[i]]));
up.samples.per.probe <- apply(is.up, 1, sum);
```

```
494
495 up.probes <- up.samples.per.probe > samples.threshold;
496
497 print(paste(sum(up.probes), "probes upregulated in at least 10% of the
    samples in",names(datalist)[[i]]));
498 updata[[i]]<-apply(is.up,2,sum);
499
500 is.down <- exprs(datalist[[i]]) < 1/fold.threshold;
501 print(paste(sum(is.down), "down-regulated probes in",names(datalist)[[i]]));
502 down.samples.per.probe <- apply(is.down, 1, sum);
503 down.probes <- down.samples.per.probe > samples.threshold;
504 print(paste(sum(down.probes), "probes downregulated in at least 10% of the
    samples in",names(datalist)[[i]]));
505 downdata[[i]]<-apply(is.down,2,sum);
506
507 is.changed <- is.up | is.down;
508 print(paste(sum(is.changed), "changed probes in",names(datalist)[[i]]));
509 changed.samples.per.probe <- apply(is.changed, 1, sum);
510 changed.probes <- changed.samples.per.probe > samples.threshold;
511 print(paste(sum(changed.probes), "probes changed in at least 10% of the
    samples in",names(datalist)[[i]]));
512 vardata<-apply(is.changed,2,sum);
513 updown.probes[[i]] <- up.probes | down.probes;
514 print(paste(sum(updown.probes[[i]]), "probes upregulated or downregulated in
     at least 10% of the samples in",names(datalist)[[i]]));
515
516 #final selection of probes passing both filters
517 selected.probes[[i]]<-present.probes[[i]]&updown.probes[[i]]
518
519 print(paste(sum(selected.probes[[i]]), "probes pass both filters in",names(
    datalist)[[i]]));
520
521 par(mar=c(7,5,1,1))
522 barplot(updata[[i]],las=2,col=as.character(tbstat[[i]]),ylab="Number of
    probes with FC > 2",cex.lab=.8,ylim=c(0,9000),main=names(datalist)[[i]])
523 export.plot(file.prefix=paste('fig',figure,'.',plotnum,names(datalist)[i],'
    probes with FC > 2'),export.formats=export.formats.plots,height=height,
    width=(1 + sqrt(5))/2*height);plotnum<-plotnum+1
524 barplot(downdata[[i]],las=2,col=as.character(tbstat[[i]]),ylab="Number of
    probes with FC< .5",cex.lab=.8,ylim=c(0,9000),main=names(datalist)[[i]])
525 export.plot(file.prefix=paste('fig',figure,'.',plotnum,names(datalist)[i],'
    probes with FC < .5'),export.formats=export.formats.plots,height=height,
    width=(1 + sqrt(5))/2*height);plotnum<-plotnum+1
526 par(parbackup)
527
528 }
529 figure<-figure+1
530
531 #visualise selected probes####
532
533 #plotnum<-1
534 #for (i in 1:length(datalist)){
535 #par(mar=c(10,7,2,1))
536 #matplot(log2(t(exprs(datalist[[i]])[selected.probes[[i]],])),type="l",ylim=
    c(-8,8),ylab="Fold Change (log2)",xaxt='n',main=paste(names(datalist)[i
    ],'profile plot selected probes'));
537 #abline(a=1,b=0,col="yellow");
538 #abline(a=-1,b=0,col="yellow");
539 #abline(a=-0,b=0,col="yellow");
```

```
#axis(1,at=1:dim(exprs(datalist[[i]]))[[2]],lab=colnames(exprs(datalist[[i]])),las=2)
#par(parbackup)

#export.plot(file.prefix=paste('fig',figure,'.',plotnum,names(datalist)[i],' profile plot selected probes'),export.formats=export.formats.plots, height=height,width=(1 + sqrt(5))/2*height);plotnum<-plotnum+1
#}

#figure<-figure+1

#heatmaps
plotnum<-1
for (i in 1:length(datalist)){
phenomatrix<-matrix(cbind(as.character(tbstat[[i]]),as.character(tbstat[[i]])),ncol=2)
superHeatmap(x=datalist[[i]],y=selected.probes[[i]],phenomatrix=phenomatrix, scale="none")
export.plot(file.prefix=paste('fig',figure,'.',plotnum,names(datalist)[i],' heatmap selected probes'),export.formats=export.formats.plots,height= height,width=(1 + sqrt(5))/2*height);plotnum<-plotnum+1
}
figure<-figure+1

#Data PCA and MDS: selected probes
plotnum<-1
for (i in 1:length(datalist)){
  plotSampleRelationsAD(datalist[[i]][selected.probes[[i]],],method="mds", color=as.character(tbstat[[i]]),plotchar=16)
  legend("bottomleft",
         list(c("Healthy","Latent","PTB","TB-PC"),c("Latent","PTB"),c("Healthy","Latent","PTB"))[[i]],
         pch=16,
         col=list(c("green","deepskyblue","orange","red"),c("deepskyblue","orange"),c("green","deepskyblue","orange"))[[i]],
         cex=.8
  )
export.plot(file.prefix=paste('fig',figure,'.',plotnum,'mds_selected',names(datalist[[i]])),export.formats=export.formats.plots,height=height*0.7, width=(1 + sqrt(5))/2*height*0.7);plotnum<-plotnum+1
}
figure<-figure+1

#selected probes
plotnum<-1
for (i in 1:length(datalist)){
  l=colnames(exprs(datalist[[i]][selected.probes[[i]],]))
  trueClasses <-list(as.factor(B.N.AX$class2),as.factor(berry.val$confirmed_diagnosis),as.factor(berry.train$confirmed_diagnosis))
  spca <- SamplePCA(exprs(datalist[[i]][selected.probes[[i]],]), trueClasses[[i]])
  plot(spca, col=list(c("deepskyblue","orange","green","red","red"),c("deepskyblue","orange"),c("green","green","deepskyblue","orange"))[[i]], main=paste(sum(selected.probes[[i]]),"selected probes",sep=" "))#,cex=2, cex.axis=2,cex.main=3,cex.lab=2.5,pch=16)
  mtext(sprintf("%.2f",spca@variances[1]/sum(spca@variances)),side=1,line=3, adj=1,cex=1.5)
  mtext(sprintf("%.2f",spca@variances[2]/sum(spca@variances)),side=2,line=3, adj=1,cex=1.5)
```

```
580   legend("bottomleft",
581          list(c("Healthy","Latent","PTB","TB-PC"),c("Latent","PTB"),c("
        Healthy","Latent","PTB"))[[i]],
582          pch=16,
583          col=list(c("green","deepskyblue","orange","red"),c("deepskyblue","
        orange"),c("green","deepskyblue","orange"))[[i]],
584          cex=.8
585   )
586   # mark the group centers
587   #x1 <- predict(spca, matrix(apply(t(exprs(datalist[[i]])[grep("b",rownames
        (exprs(datalist[[i]]))),]), 1, mean), ncol=1))
588   #points(x1[1], x1[2], col='red', cex=6,pch=9)
589   #x2 <- predict(spca, matrix(apply(t(exprs(datalist[[i]])[grep("f",rownames
        (exprs(datalist[[i]]))),]), 1, mean), ncol=1))
590   #points(x2[1], x2[2], col='blue', cex=6,pch=9)
591   export.plot(file.prefix=paste('fig',figure,'.',plotnum,'pca_selected',
        names(datalist[[i]])),export.formats=export.formats.plots,height=height*
        0.7,width=(1 + sqrt(5))/2*height*0.7);plotnum<-plotnum+1
592 }
593 figure<-figure+1
594
595 #statistics####
596
597 #classes
598
599 tbstat.class=list(as.factor(B.N.AX$class2),as.factor(berry.val$confirmed_
        diagnosis),as.factor(berry.train$confirmed_diagnosis))
600
601 ##this needs to be sorted out!
602 #this depends on the order of samples in the RData file
603 #consider automating this instead of manually defining
604 levels(tbstat.class[[1]])<-c("no TB","no TB","active TB","active TB","active
         TB")#force 2 condition KW
605 levels(tbstat.class[[2]])<-c("no TB","active TB")
606 levels(tbstat.class[[3]])<-c("no TB","no TB","latent TB","active TB")#3
        condition KW as per Nature paper
607 #levels(tbstat.class[[3]])<-c("no TB","no TB","no TB","active TB")#2
        condition KW as per Nature paper
608
609 data.fil=list()
610 for (i in 1:length(datalist)){
611
612 data.fil[[i]]<-datalist[[i]][selected.probes[[i]],]
613 par(mar=c(7,5,1,1))
614 boxplot(data.fil[[i]],las=2,col=as.character(tbstat[[i]]),main=paste(names(
        datalist)[[i]],'Boxplot of selected probes'))
615 par(parbackup)
616 export.plot(file.prefix=paste('fig',figure,'.',i,names(datalist)[[i]],'
        boxplot selected probes'),export.formats=export.formats.plots,height=
        height,width=(1 + sqrt(5))/2*height)
617 }
618 figure<-figure+1
619
620
621 #KW for loop
622 data.fil.probe.names<-list()
623 data.fil.probenumber<-list()
624 for (i in 1:length(datalist)){
625 data.fil.probe.names[[i]]<-rownames(exprs(data.fil[[i]]))
```

```
626 data.fil.probenumber[[i]]<-nrow(data.fil[[i]])
627 }
628
629
630 #Let's first set alpha:
631 #kruskal.wallis.alpha <- 0.01 nil significant here... and limma also set to
        0.05, so it's only fair
632 kruskal.wallis.alpha <- 0.01
633
634 kruskal.wallis.table<-list(data.frame(),data.frame(),data.frame())
635
636 ## Run the KW test on gene
637     for (i in 1:length(datalist)){
638       ksdat<-as.matrix(exprs(data.fil[[i]]))
639       g=tbstat.class[[i]]
640       for(j in 1:data.fil.probenumber[[i]]){
641           x <- as.vector(ksdat[j,])
642           ks.test <- kruskal.test(x, g)
643           ## Store the result in the data frame
644           kruskal.wallis.table[[i]] <- rbind(kruskal.wallis.table[[i]],data.
        frame(id=data.fil.probe.names[[i]][j],p.value=ks.test$p.value))
645       }
646     }
647
648 nb.tests<-list()
649 for (i in 1:length(datalist)){
650 nb.tests[[i]] <- data.fil.probenumber[[i]]
651 kruskal.wallis.table[[i]]$E.value <- kruskal.wallis.table[[i]]$p.value * nb.
        tests[[i]]
652 kruskal.wallis.table[[i]]$FWER <- pbinom(q=0, p=kruskal.wallis.table[[i]]$p.
        value,size=nb.tests[[i]], lower.tail=FALSE)
653 kruskal.wallis.table[[i]] <- kruskal.wallis.table[[i]][order(kruskal.wallis.
        table[[i]]$p.value,decreasing=FALSE), ]
654 kruskal.wallis.table[[i]]$q.value.factor <- nb.tests[[i]] / 1:nb.tests[[i]]
655 kruskal.wallis.table[[i]]$q.value <- kruskal.wallis.table[[i]]$p.value *
        kruskal.wallis.table[[i]]$q.value.factor
656 }
657
658 head(kruskal.wallis.table[[1]])
659 head(kruskal.wallis.table[[2]])
660 head(kruskal.wallis.table[[3]])
661
662
663 for (i in 1:length(datalist)){
664
665 plot(kruskal.wallis.table[[i]]$p.value,
666      kruskal.wallis.table[[i]]$E.value,
667      main=paste('Multitesting corrections: ',names(datalist)[i]),
668      xlab='Nominal p-value',
669      ylab='Multitesting-corrected statistics',
670      log='xy',
671      col='blue',
672      panel.first=grid(col='#BBBBBB',lty='solid'))
673 lines(kruskal.wallis.table[[i]]$p.value,
674       kruskal.wallis.table[[i]]$FWER,
675       pch=20,col='darkgreen', type='p'
676 )
677 lines(kruskal.wallis.table[[i]]$p.value,
678       kruskal.wallis.table[[i]]$q.value,
```

```
679       pch='+',col='darkred', type='p'
680 )
681 abline(h=kruskal.wallis.alpha, col='red', lwd=2)
682 legend('topleft', legend=c('E-value', 'p-value', 'q-value'), col=c('blue', '
    darkgreen',
683                                                                    'darkred'
    ), lwd=2,bg='white',bty='o')
684 }
685
686 #results####
687 #calculation of effect size for original nature paper
688 effect.size<-pwr.t2n.test(n1=13,n2=29,d=NULL,sig.level=0.01,power=0.9,
    alternative="two.sided")$d
689
690 #calculation of significance level, using effect size calculated above with
    90% power for t-test #note this does not work when doing stats using
    Kruskal Wallis ANOVA for 3 conditions...
691
692 #sig<-list(
693 #  pwr.t2n.test(n1=12,n2=10,d=effect.size,sig.level=NULL,power=0.9,
    alternative="two.sided")$sig.level,
694 #  pwr.t2n.test(n1=20,n2=31,d=effect.size,sig.level=NULL,power=0.9,
    alternative="two.sided")$sig.level,
695 #  pwr.t2n.test(n1=13,n2=29,d=effect.size,sig.level=NULL,power=0.9,
    alternative="two.sided")$sig.level
696 #)
697
698 #alt sig - empiric
699 sig<-list(0.08,0.0001,0.01)
700
701 diff.genes<-list()
702 for (i in 1:length(datalist)){
703 last.significant.element <- max(which(kruskal.wallis.table[[i]]$q.value <=
    sig[[i]]))
704 selected <- 1:last.significant.element
705 diff.genes.factor <- kruskal.wallis.table[[i]]$id[selected]
706 diff.genes[[i]] <- as.vector(diff.genes.factor)
707 }
708
709 intersect12<-intersect(diff.genes[[1]],diff.genes[[2]])
710 intersect13<-intersect(diff.genes[[1]],diff.genes[[3]])
711 intersect23<-intersect(diff.genes[[2]],diff.genes[[3]])
712
713 #this can be compared to limma, also at 0.05 given sample size
714
715 #heatmaps####
716 plotnum<-1
717 for (i in 1:length(datalist)){
718   phenomatrix<-matrix(cbind(as.character(tbstat[[i]]),as.character(tbstat[[i
    ]])),ncol=2)
719   superHeatmap(x=datalist[[i]],y=diff.genes[[i]],phenomatrix=phenomatrix,
    scale="none",addTit=paste("diff. exp.",names(datalist)[i]))
720   export.plot(file.prefix=paste('fig',figure,'.',plotnum,names(datalist)[i],
    'heatmap DE probes'),export.formats=export.formats.plots,height=height,
    width=(1 + sqrt(5))/2*height);plotnum<-plotnum+1
721 }
722 figure<-figure+1
723
724 #Two-way overlap heatmaps not run####
```

```
# plotnum<-1
# for (i in 1:length(datalist)){
#   phenomatrix<-matrix(cbind(as.character(tbstat[[i]]),as.character(tbstat[[i]])),ncol=2)
#   superHeatmap(x=datalist[[i]],y=intersect12,phenomatrix=phenomatrix,scale="none",addTit=paste("overlap IMPI VAL",names(datalist)[i]))
#   export.plot(file.prefix=paste('fig',figure,'.',plotnum,names(datalist)[i],'heatmap DE probes: overlap IMPI VAL'),export.formats=export.formats.plots,height=height,width=(1 + sqrt(5))/2*height);plotnum<-plotnum+1
#   }
# figure<-figure+1
#
# plotnum<-1
# for (i in 1:length(datalist)){
#   phenomatrix<-matrix(cbind(as.character(tbstat[[i]]),as.character(tbstat[[i]])),ncol=2)
#   superHeatmap(x=datalist[[i]],y=intersect13,phenomatrix=phenomatrix,scale="none",addTit=paste("overlap IMPI TRAIN",names(datalist)[i]))
#   export.plot(file.prefix=paste('fig',figure,'.',plotnum,names(datalist)[i],'heatmap DE probes: overlap IMPI TRAIN'),export.formats=export.formats.plots,height=height,width=(1 + sqrt(5))/2*height);plotnum<-plotnum+1
# }
# figure<-figure+1
#
#
# #the below gives an interesting error using intersect23: intersect23[25] does not exist in datalist[[1]]; it's a nuID not found on newer chips
# #fix:
# intersect23b<-intersect(row.names(exprs(datalist[[1]])),intersect23)
#
# plotnum<-1
# for (i in 1:length(datalist)){
#   phenomatrix<-matrix(cbind(as.character(tbstat[[i]]),as.character(tbstat[[i]])),ncol=2)
#   superHeatmap(x=datalist[[i]],y=intersect23b,phenomatrix=phenomatrix,scale="none",addTit=paste("overlap VAL TRAIN",names(datalist)[i]))
#   export.plot(file.prefix=paste('fig',figure,'.',plotnum,names(datalist)[i],'heatmap DE probes: overlap VAL TRAIN'),export.formats=export.formats.plots,height=height,width=(1 + sqrt(5))/2*height);plotnum<-plotnum+1
# }
# figure<-figure+1

#Data PCA and MDS####

plotnum<-1
for (i in 1:length(datalist)){
  plotSampleRelationsAD(datalist[[i]][diff.genes[[i]],],method="mds",color=as.character(tbstat[[i]]),plotchar=16)
  legend("bottomleft",
         list(c("Healthy","Latent","PTB","TB-PC"),c("Latent","PTB"),c("Healthy","Latent","PTB"))[[i]],
         pch=16,
         col=list(c("green","deepskyblue","orange","red"),c("deepskyblue","orange"),c("green","deepskyblue","orange"))[[i]],
         cex=.8
  )
export.plot(file.prefix=paste('fig',figure,'.',plotnum,'mds_selected',names(datalist[[i]])),export.formats=export.formats.plots,height=height*0.7,width=(1 + sqrt(5))/2*height*0.7);plotnum<-plotnum+1
```

```
}
figure<-figure+1

#selected genes
plotnum<-1
for (i in 1:length(datalist)){
  l=colnames(exprs(datalist[[i]][diff.genes[[i]],]))
  trueClasses <-list(as.factor(B.N.AX$class2),as.factor(berry.val$confirmed_diagnosis),as.factor(berry.train$confirmed_diagnosis))
  spca <- SamplePCA(exprs(datalist[[i]][diff.genes[[i]],]), trueClasses[[i]])
  plot(spca, col=list(c("deepskyblue","orange","green","red","red"),c("deepskyblue","orange"),c("green","green","deepskyblue","orange"))[[i]], main=paste(length(diff.genes[[i]]),"selected probes",sep=" "))#,cex=2,cex.axis=2,cex.main=3,cex.lab=2.5,pch=16)
  mtext(sprintf("%.2f",spca@variances[1]/sum(spca@variances)),side=1,line=3,adj=1,cex=1.5)
  mtext(sprintf("%.2f",spca@variances[2]/sum(spca@variances)),side=2,line=3,adj=1,cex=1.5)
  legend("bottomleft",
         list(c("Healthy","Latent","PTB","TB-PC"),c("Latent","PTB"),c("Healthy","Latent","PTB"))[[i]],
         pch=16,
         col=list(c("green","deepskyblue","orange","red"),c("deepskyblue","orange"),c("green","deepskyblue","orange"))[[i]],
         cex=.8
  )
  # mark the group centers
  #x1 <- predict(spca, matrix(apply(t(exprs(datalist[[i]])[grep("b",rownames(exprs(datalist[[i]]))),]), 1, mean), ncol=1))
  #points(x1[1], x1[2], col='red', cex=6,pch=9)
  #x2 <- predict(spca, matrix(apply(t(exprs(datalist[[i]])[grep("f",rownames(exprs(datalist[[i]]))),]), 1, mean), ncol=1))
  #points(x2[1], x2[2], col='blue', cex=6,pch=9)
  export.plot(file.prefix=paste('fig',figure,'.',plotnum,'pca_selected',names(datalist[[i]])),export.formats=export.formats.plots,height=height*0.7,width=(1 + sqrt(5))/2*height*0.7);plotnum<-plotnum+1
}
figure<-figure+1
##

#Export the probe lists####
#######################################.

dict.bnax<-nuID2IlluminaID(as.character(diff.genes[[1]]),species="Human")
dict.val<-nuID2IlluminaID(as.character(diff.genes[[2]]),species="Human")
dict.train<-nuID2IlluminaID(as.character(diff.genes[[3]]),species="Human")

dict.bnax.entrez<-nuID2EntrezID(as.character(diff.genes[[1]]),filterTh = NULL,lib.mapping='lumiHumanIDMapping', returnAllInfo = TRUE)
dict.val.entrez<-nuID2EntrezID(as.character(diff.genes[[2]]),filterTh = NULL,lib.mapping='lumiHumanIDMapping', returnAllInfo = TRUE)
dict.train.entrez<-nuID2EntrezID(as.character(diff.genes[[3]]),filterTh = NULL,lib.mapping='lumiHumanIDMapping', returnAllInfo = TRUE)

dict2.bnax<-merge(dict.bnax,kruskal.wallis.table[[1]],by.x="nuID",by.y="id")
dict2.val<-merge(dict.val,kruskal.wallis.table[[2]],by.x="nuID",by.y="id")
dict2.train<-merge(dict.train,kruskal.wallis.table[[3]],by.x="nuID",by.y="id")
```

```
808
809 dict3.bnax<-merge(dict2.bnax,dict.bnax.entrez,by.x="nuID",by.y=0)
810 dict3.val<-merge(dict2.val,dict.val.entrez,by.x="nuID",by.y=0)
811 dict3.train<-merge(dict2.train,dict.train.entrez,by.x="nuID",by.y=0)
812
813 write.csv(dict3.bnax,file=file.path(dir.results,"DE_Baylor_BNAX.csv"))
814 write.csv(dict3.val,file=file.path(dir.results,"DE_Baylor_val.csv"))
815 write.csv(dict3.train,file=file.path(dir.results,"DE_Baylor_train.csv"))
816
817 ########################################
818 #overlap computations for 393
819 ########################################
820
821 #import probe lists
822
823 #load Berry 393 and 86 probe data####
824
825 load(file.path(dir.rdata,'berry.sig.RData'))
826 load(file.path(dir.rdata,'berry.86.RData'))
827 load(file.path(dir.rdata,'berry.393.RData'))
828
829 #annotate probe lists
830 dict<-probeID2nuID(berry.sig,species="Human")
831 dict.86<-IlluminaID2nuID(as.character(berry.86[,1]),species="Human")
832 dict.393<-IlluminaID2nuID(as.character(berry.393[,1]),species="Human")
833
834 #intersectOfOverlapWith393<-intersect(intersect13,dict[,7])
835
836 #intersectBerryWith393<-intersect(diff.genes[[2]],dict[,7])
837
838 #intersectSAWith393<-intersect(diff.genes[[1]],dict[,7])
839
840 ##Venn 4-way at probe level####
841 x.1<-diff.genes[[1]]
842 x.2<-diff.genes[[2]]
843 x.3<-diff.genes[[3]]
844 x.4<-as.character(dict[,7])
845 x.5<-as.character(dict.86[,7])
846
847 venndatax.hivNeg<-list(x.1,x.2,x.3,x.4)
848 sixtyone<-intersect(intersect(intersect(x.1,x.2),x.3),x.4)
849 so.a<-nuID2IlluminaID(as.character(sixtyone),species="Human")
850 so.b<-nuID2EntrezID(as.character(sixtyone),filterTh = NULL,lib.mapping='
      lumiHumanIDMapping', returnAllInfo = TRUE)
851 so.c<-merge(so.a,so.b,by.x="nuID",by.y=0)
852
853 write.csv(so.c,file=file.path(dir.results,"TripleOverlap61probes.csv"))
854
855 temp.venn<-venn.diagram(
856   x = venndatax.hivNeg,
857   category=c("SA","VAL","TRAIN","393"),
858   #filename = file.path(dir.figures,"fig71_Venn_3set_lh_al_ah.tiff"),
859   #filename = file.path(dir.figures,paste("fig",figure,"_Venn_4set_probe.
       tiff",sep="")),
860   filename = NULL,
861   #filename = "/Users/armindeffur/Desktop/file.tiff",
862   scaled = T,  ext.text = TRUE,  ext.line.lwd = 2,
863   ext.dist = -0.15,  ext.length = 0.9,  ext.pos = -4,
864   inverted = TRUE,
```

```
  cex = 2.5,  cat.cex = 2.5,  rotation.degree = 0,
  #main = "Overlap",  sub = "MA, VAL, TRAIN","393",
  # main.cex = 2,  sub.cex = 1,
  fill=c("blue","green","red","gray"),
  alpha=c(.4,.4,.4,.4),height=3000,width=(1 + sqrt(5))/2*3000
)

pdf(file=file.path(dir.figures,paste("fig",figure,"_Venn_4set_probe.pdf",sep
    ="")),height=7,width=(1 + sqrt(5))/2*7)
grid.draw(temp.venn)
dev.off()
figure<-figure+1

##Venn 5-way at probe level####
x.1<-diff.genes[[1]]
x.2<-diff.genes[[2]]
x.3<-diff.genes[[3]]
x.4<-as.character(dict[,7])
x.5<-as.character(dict.86[,7])

venndatax.hivNeg<-list(x.1,x.2,x.3,x.4,x.5)

temp.venn<-venn.diagram(
  x = venndatax.hivNeg,
  category=c("SA","VAL","TRAIN","393","86"),
  #filename = file.path(dir.figures,"fig71_Venn_3set_lh_al_ah.tiff"),
  #filename = file.path(dir.figures,paste("fig",figure,"_Venn_4set_probe.
    tiff",sep="")),
  filename = NULL,
  #filename = "/Users/armindeffur/Desktop/file.tiff",
  scaled = T,  ext.text = TRUE,  ext.line.lwd = 2,
  ext.dist = -0.15,  ext.length = 0.9,  ext.pos = -4,
  inverted = TRUE,
  cex = 2.5,  cat.cex = 2.5,  rotation.degree = 0,
  #main = "Overlap",  sub = "MA, VAL, TRAIN","393",
  # main.cex = 2,  sub.cex = 1,
  fill=c("blue","green","red","gray","orange"),
  alpha=c(.4,.4,.4,.4,.4),height=3000,width=(1 + sqrt(5))/2*3000
)

pdf(file=file.path(dir.figures,paste("fig",figure,"_Venn_4set_probe.pdf",sep
    ="")),height=7,width=(1 + sqrt(5))/2*7)
grid.draw(temp.venn)
dev.off()
figure<-figure+1

##Venn 4-way at gene level####
gx.1<-unique(as.character(nuID2IlluminaID(x.1)[,2]))
gx.2<-unique(as.character(nuID2IlluminaID(x.2)[,2]))
gx.3<-unique(as.character(nuID2IlluminaID(x.3)[,2]))
gx.4<-unique(as.character(nuID2IlluminaID(x.4)[,2]))

venndatax.hivNeg.genes<-list(gx.1,gx.2,gx.3,gx.4)

temp.venn.2<-venn.diagram(
  x = venndatax.hivNeg.genes,
  category=c("SA","VAL","TRAIN","393"),
  #filename = file.path(dir.figures,"fig71_Venn_3set_lh_al_ah.tiff"),
```

```
  #filename = file.path(dir.figures,paste("fig",figure,"_Venn_4set_gene.tiff
    ",sep="")),
  filename = NULL,
  scaled = T,  ext.text = TRUE,  ext.line.lwd = 2,
  ext.dist = -0.15,  ext.length = 0.9,  ext.pos = -4,
  inverted = TRUE,
  cex = 2.5,  cat.cex = 2.5,  rotation.degree = 0,
  #main = "Overlap",  sub = "MA, VAL, TRAIN","393",
  # main.cex = 2,  sub.cex = 1,
  fill=c("blue","green","red","gray"),
  alpha=c(.4,.4,.4,.4),height=3000,width=(1 + sqrt(5))/2*3000
)
pdf(file=file.path(dir.figures,paste("fig",figure,"_Venn_4set_probe.pdf",sep
    "")),height=7,width=(1 + sqrt(5))/2*7)
grid.draw(temp.venn.2)
dev.off()
figure<-figure+1

##Venn 4-way at probe level for top 500 probes####

xEQ.1<-as.character(kruskal.wallis.table[[1]][1:500,1])
xEQ.2<-as.character(kruskal.wallis.table[[2]][1:500,1])
xEQ.3<-as.character(kruskal.wallis.table[[3]][1:500,1])
x.4<-as.character(dict[,7]);
venndatax.hivNeg.500<-list(xEQ.1,xEQ.2,xEQ.3,x.4)
temp.venn.3<-venn.diagram(
  x = venndatax.hivNeg.500,
  category=c("SA","VAL","TRAIN","393"),
  #filename = file.path(dir.figures,"fig71_Venn_3set_lh_al_ah.tiff"),
  #filename = file.path(dir.figures,paste("fig",figure,"_Venn_4set_top_500.
    tiff",sep="")),
  filename = NULL,
  scaled = T,  ext.text = TRUE,  ext.line.lwd = 2,
  ext.dist = -0.15,  ext.length = 0.9,  ext.pos = -4,
  inverted = TRUE,
  cex = 2.5,  cat.cex = 2.5,  rotation.degree = 0,
  #main = "Overlap",  sub = "MA, VAL, TRAIN","393",
  # main.cex = 2,  sub.cex = 1,
  fill=c("blue","green","red","gray"),
  alpha=c(.4,.4,.4,.4),height=3000,width=(1 + sqrt(5))/2*3000
)
pdf(file=file.path(dir.figures,paste("fig",figure,"_Venn_4set_probe.pdf",sep
    "")),height=7,width=(1 + sqrt(5))/2*7)
grid.draw(temp.venn.3)
dev.off()
figure<-figure+1

##Venn 4-way at gene level for top 500 probes####

g500x.1<-unique(as.character(nuID2IlluminaID(xEQ.1)[,2]))
g500x.2<-unique(as.character(nuID2IlluminaID(xEQ.2)[,2]))
g500x.3<-unique(as.character(nuID2IlluminaID(xEQ.3)[,2]))
g500x.4<-unique(as.character(nuID2IlluminaID(x.4)[,2]))

venndatax.hivNeg.genes.500<-list(g500x.1,g500x.2,g500x.3,g500x.4)

temp.venn.4<-venn.diagram(
  x = venndatax.hivNeg.genes.500,
  category=c("SA","VAL","TRAIN","393"),
```

```
975   #filename = file.path(dir.figures,"fig71_Venn_3set_lh_al_ah.tiff"),
976   #filename = file.path(dir.figures,paste("fig",figure,"_Venn_4set_gene.tiff
        ",sep="")),
977   filename = NULL,
978   scaled = T,  ext.text = TRUE,  ext.line.lwd = 2,
979   ext.dist = -0.15,  ext.length = 0.9,  ext.pos = -4,
980   inverted = TRUE,
981   cex = 2.5,  cat.cex = 2.5,  rotation.degree = 0,
982   #main = "Overlap",  sub = "MA, VAL, TRAIN","393",
983   # main.cex = 2,  sub.cex = 1,
984   fill=c("blue","green","red","gray"),
985   alpha=c(.4,.4,.4,.4),height=3000,width=(1 + sqrt(5))/2*3000
986 )
987 pdf(file=file.path(dir.figures,paste("fig",figure,"_Venn_4set_probe.pdf",sep
      ="")),height=7,width=(1 + sqrt(5))/2*7)
988 grid.draw(temp.venn.4)
989 dev.off()
990 figure<-figure+1
991
992 ##Heatmaps: full intersect at probe and gene level####
993
994 in.common.probe<-intersect(x.1,intersect(x.2,intersect(x.3,x.4)))
995 in.common<-intersect(gx.1,intersect(gx.2,intersect(gx.3,gx.4)))
996
997 plotnum<-1
998 for (i in 1:length(datalist)){
999   phenomatrix<-matrix(cbind(as.character(tbstat[[i]]),as.character(tbstat[[i
        ]])),ncol=2)
1000  superHeatmap(x=datalist[[i]],y=in.common.probe,phenomatrix=phenomatrix,
        scale="none",addTit=paste("overlap",names(datalist)[i]))
1001  export.plot(file.prefix=paste('fig',figure,'.',plotnum,names(datalist)[i],
        'heatmap in common probes'),export.formats=export.formats.plots,height=
        height,width=(1 + sqrt(5))/2*height);plotnum<-plotnum+1
1002 }
1003 figure<-figure+1
1004
1005 ##Venn: 393 and AOG_R_method_train probe (validation of implementation of
      Baylor method in R)#####
1006 venndatax.val.meth<-list(x.3,x.4)
1007 temp.venn.5<-venn.diagram(
1008   x = venndatax.val.meth,
1009   category=c("TRAIN","393"),
1010   #filename = file.path(dir.figures,"fig71_Venn_3set_lh_al_ah.tiff"),
1011   #filename = file.path(dir.figures,paste("fig",figure,"_Venn_4set_gene.tiff
        ",sep="")),
1012   filename = NULL,
1013   scaled = F,  ext.text = TRUE,  ext.line.lwd = 2,
1014   ext.dist = -0.15,  ext.length = 0.9,  ext.pos = -4,
1015   inverted = TRUE,
1016   cex = 2.5,  cat.cex = 2.5,  rotation.degree = 0,
1017   #main = "Overlap",  sub = "MA, VAL, TRAIN","393",
1018   # main.cex = 2,  sub.cex = 1,
1019   fill=c("blue","green"),
1020   alpha=c(.4,.4),height=3000,width=(1 + sqrt(5))/2*3000
1021 )
1022 pdf(file=file.path(dir.figures,paste("fig",figure,"_Venn_2set_probe.pdf",sep
      ="")),height=7,width=(1 + sqrt(5))/2*7)
1023 grid.draw(temp.venn.5)
1024 dev.off()
```

```
figure<-figure+1

##Venn: 393 and AOG_R_method_train gene (validation of implementation of
    Baylor method in R) #####
venndatax.val.meth.g<-list(gx.3,gx.4)
temp.venn.6<-venn.diagram(
  x = venndatax.val.meth.g,
  category=c("TRAIN","393"),
  #filename = file.path(dir.figures,"fig71_Venn_3set_lh_al_ah.tiff"),
  #filename = file.path(dir.figures,paste("fig",figure,"_Venn_4set_gene.tiff
    ",sep="")),
  filename = NULL,
  scaled = F,  ext.text = TRUE,  ext.line.lwd = 2,
  ext.dist = -0.15,  ext.length = 0.9,  ext.pos = -4,
  inverted = TRUE,
  cex = 2.5,  cat.cex = 2.5,  rotation.degree = 0,
  #main = "Overlap",  sub = "MA, VAL, TRAIN","393",
  # main.cex = 2,  sub.cex = 1,
  fill=c("blue","green"),
  alpha=c(.4,.4),height=3000,width=(1 + sqrt(5))/2*3000
)
pdf(file=file.path(dir.figures,paste("fig",figure,"_Venn_2set_probe.pdf",sep
    ="")),height=7,width=(1 + sqrt(5))/2*7)
grid.draw(temp.venn.6)
dev.off()
figure<-figure+1

##Venn: SA and 393 probe ####

venndatax.val.SA<-list(x.1,x.4)
temp.venn.7<-venn.diagram(
  x = venndatax.val.SA,
  category=c("SA","393"),
  #filename = file.path(dir.figures,"fig71_Venn_3set_lh_al_ah.tiff"),
  #filename = file.path(dir.figures,paste("fig",figure,"_Venn_4set_gene.tiff
    ",sep="")),
  filename = NULL,
  scaled = F,  ext.text = TRUE,  ext.line.lwd = 2,
  ext.dist = -0.15,  ext.length = 0.9,  ext.pos = -4,
  inverted = TRUE,
  cex = 2.5,  cat.cex = 2.5,  rotation.degree = 0,
  #main = "Overlap",  sub = "MA, VAL, TRAIN","393",
  # main.cex = 2,  sub.cex = 1,
  fill=c("blue","green"),
  alpha=c(.4,.4),height=3000,width=(1 + sqrt(5))/2*3000
)
pdf(file=file.path(dir.figures,paste("fig",figure,"_Venn_2set_probe.pdf",sep
    ="")),height=7,width=(1 + sqrt(5))/2*7)
grid.draw(temp.venn.7)
dev.off()
figure<-figure+1

##Venn: SA and 393 gene ####

venndatax.val.SA.g<-list(gx.1,gx.4)
temp.venn.8<-venn.diagram(
  x = venndatax.val.SA.g,
  category=c("SA","393"),
  #filename = file.path(dir.figures,"fig71_Venn_3set_lh_al_ah.tiff"),
```

```
1079   #filename = file.path(dir.figures,paste("fig",figure,"_Venn_4set_gene.tiff
         ",sep="")),
1080   filename = NULL,
1081   scaled = T,  ext.text = TRUE,  ext.line.lwd = 2,
1082   ext.dist = -0.15,  ext.length = 0.9,  ext.pos = -4,
1083   inverted = TRUE,
1084   cex = 2.5,  cat.cex = 2.5,  rotation.degree = 0,
1085   #main = "Overlap",  sub = "MA, VAL, TRAIN","393",
1086   # main.cex = 2,  sub.cex = 1,
1087   fill=c("blue","green"),
1088   alpha=c(.4,.4),height=3000,width=(1 + sqrt(5))/2*3000
1089 )
1090 pdf(file=file.path(dir.figures,paste("fig",figure,"_Venn_2set_probe.pdf",sep
       =="")),height=7,width=(1 + sqrt(5))/2*7)
1091 grid.draw(temp.venn.8)
1092 dev.off()
1093 figure<-figure+1
```

Listing A.9: RelationIMPI-MA to TB-AOG

A.6 Overview of all data

```
#setup####
source("/Users/armindeffur/Documents/002_Science_universal/Active_Research/IMPI-Microarray/Projects/01_scripts/06_IMPI-MA/Analysis/00_System/System_setup.R")
source(file.path(dir.R.files,"superHeatmap.R"))
par("xpd"=FALSE)
export.formats.plots <- c("pdf","png")
#phenodata directory
dir.pheno <- file.path(dir.data_root, '09_IMPI_MA_DATA/08_PhenoData')
print(paste("Phenodata data repository", dir.pheno))

## Individual data folders (specific for each part)
dir.rdata <- file.path(dir.data_root,'09_IMPI_MA_DATA/05_RData')

#Output folders
dir.home<-dir.root
dir.main <- file.path(dir.root, '02_output/06_IMPI-MA/AllData','QC_and_Unsupervised')

#Current date string used for versioning output
ds<-date()

#versioned output
dir.output.version<-file.path(dir.main,paste("version_",ds))

## Define folder for storing results NB do this for each analysis
dir.results <- file.path(dir.output.version, "results")
## Define folder for saving figures
dir.figures <- file.path(dir.output.version, "figures")
for (dir in c(dir.main, dir.figures, dir.results)) {
  if (!file.exists(dir)) {
    dir.create(dir, recursive=T,showWarnings=T)
  }
}

sink(type="output",file=file.path(dir.results,"session_output.txt"))
sessionInfo()
print(paste("Begin time:",ds))

print(paste("Results will be saved to", dir.results))
print(paste("Figures will be saved to", dir.figures))


figure<-1

#data
load(file.path(dir.rdata,'IMPI_MA.RData'))
load(file.path(dir.rdata,'berry_train.RData'))
load(file.path(dir.rdata,'berry_test.RData'))
load(file.path(dir.rdata,'berry_val.RData'))

#subset allArrays by compartment
fluidArrays<-allArrays[,allArrays$Compartment=="fluid"]
bloodArrays<-allArrays[,allArrays$Compartment=="blood"]

#treemap####
```

```
54
55 datacats<-read.csv(file.path(dir.pheno,"phenodf_simple.csv"))
56
57 tmPlot(datacats,index=c("class2","Compartment","matching","HIV.Status"),
       vSize="N",vColor="N",type="value",pal="Set3")
58 tmPlot(datacats,index=c("class2","Compartment","HIV.Status"),vSize="N",
       vColor="N",type="value",pal="Set3")
59
60 #QC and plots####
61
62 #QC summary
63 allarraysum<-summary(allArrays,"QC")
64 allArrays
65
66 berry.trainsum<-summary(berry.train,"QC")
67 berry.train
68
69 berry.testsum<-summary(berry.test,"QC")
70 berry.train
71
72 berry.valsum<-summary(berry.val,"QC")
73 berry.val
74
75 par(mfrow=c(1,1),mar=c(5, 4, 4, 2) + 0.1)
76 setwd(dir.figures)
77
78 array.list<-list("All IMPI Arrays"=allArrays,"IMPI blood Arrays"=bloodArrays
       ,"IMPI fluid Arrays"=fluidArrays,"Berry train"=berry.train,"Berry test"=
       berry.test,"Berry SA validation"=berry.val)
79
80 # #PDF plots
81 # plotnumber<-1
82 # for (i in 1:length(array.list)) {
83 #   density(array.list[[i]],addLegend=FALSE,main=paste("Density plot",names(
       array.list)[i]))
84 #   export.plot(file.prefix=paste('fig',figure,'.',plotnumber,names(array.
       list[i])),export.formats=export.formats.plots,height=height/2,width=(1 +
        sqrt(5))/2*height/2)
85 #   plotnumber<-plotnumber+1
86 # }
87 # figure<-figure+1
88 #
89 # #CDF plots
90 # plotnumber<-1
91 # for (i in 1:length(array.list)) {
92 #   plotCDF(array.list[[i]],addLegend=FALSE,main=paste("CDF plot",names(
       array.list)[i]))
93 #   export.plot(file.prefix=paste('fig',figure,'.',plotnumber,names(array.
       list[i])),export.formats=export.formats.plots,height=height/2,width=(1 +
        sqrt(5))/2*height/2)
94 #   plotnumber<-plotnumber+1
95 # }
96 # figure<-figure+1
97
98
99 #thesis versions:
100 #PDF plots
101 plotnumber<-1
102 for (i in 1:length(array.list)) {
```

```
103   density(array.list[[i]],addLegend=FALSE,main=NULL)
104   export.plot(file.prefix=paste('fig',figure,'.',plotnumber,names(array.list
        [i])),export.formats=export.formats.plots,height=height/1.5,width=(1 +
        sqrt(5))/2*height/1.5)
105   plotnumber<-plotnumber+1
106 }
107 figure<-figure+1
108
109 #CDF plots
110 plotnumber<-1
111 for (i in 1:length(array.list)) {
112   plotCDF(array.list[[i]],addLegend=FALSE,main=NULL)
113   export.plot(file.prefix=paste('fig',figure,'.',plotnumber,names(array.list
        [i])),export.formats=export.formats.plots,height=height/1.5,width=(1 +
        sqrt(5))/2*height/1.5)
114   plotnumber<-plotnumber+1
115 }
116 figure<-figure+1
117
118 #example of a pairplot of 5 samples
119 pairs(allArrays[,1:5],smoothScatter=TRUE,main="Smooth scatter, arrays 1 to 5
      ")
120 export.plot(file.prefix=paste('fig',figure,'smooth_scatter'),export.formats=
      export.formats.plots,height=height,width=(1 + sqrt(5))/2*height)
121 figure<-figure+1
122
123 #examples of MA plot of 5 samples
124 MAplot(allArrays[,1:5], smoothScatter=TRUE,main="MA plot, arrays 1 to 5")
125 export.plot(file.prefix=paste('fig',figure,'MA_1_to_5'),export.formats=
      export.formats.plots,height=height,width=(1 + sqrt(5))/2*height)
126 figure<-figure+1
127
128 #colour factors for labeling plots by compartment
129 comlist<-as.factor(allArrays$Compartment)
130 levels(comlist)<-c("gray","yellow")
131 comlist2<-comlist
132 levels(comlist2)<-c("red","tan1")
133 comlist3<-as.factor(allArrays$class2)
134 levels(comlist3)<-c("deepskyblue","orange","orange","green","red","red","red
      ")
135 pchlist<-comlist
136 levels(pchlist)<-c(21,24)
137 pchlist2<-as.numeric(as.vector(pchlist))
138
139 levels(pchlist)<-c(16,1)
140 pchlist3<-as.numeric(as.vector(pchlist))
141
142 #boxplots of all IMPI data
143 boxplot(allArrays,col=as.character(comlist),subset=NULL,main="Boxplot of all
       samples",boxwex=.8,cex=2)
144 export.plot(file.prefix=paste('fig',figure,'boxplot_all_IMPI'),export.
      formats=export.formats.plots,height=height*1.0,width=(1 + sqrt(5))/2*
      height*1.0)
145 figure<-figure+1
146
147 #Sample relations of all IMPI_MA data####
148
149 labs<-allArrays$sample_name
150 plotSampleRelation(allArrays,method="mds",color=as.character(comlist))
```

```
151 export.plot(file.prefix=paste('fig',figure,'mds_all_IMPI'),export.formats=
        export.formats.plots,height=height,width=(1 + sqrt(5))/2*height)
152 figure<-figure+1
153
154 plotSampleRelation(allArrays,method="cluster",labels=labs)
155 export.plot(file.prefix=paste('fig',figure,'sample_hc'),export.formats=
        export.formats.plots,height=height,width=(1 + sqrt(5))/2*height)
156 figure<-figure+1
157
158 plotSampleRelationsAD(allArrays,method="mds",color=as.character(comlist),
        backgr=as.character(comlist2))
159 export.plot(file.prefix=paste('fig',figure,'mds_all_IMPI_v2'),export.formats
        =export.formats.plots,height=height,width=(1 + sqrt(5))/2*height)
160 figure<-figure+1
161
162 plotSampleRelationsAD(allArrays,subset=NULL,method="mds")
163
164 plotSampleRelationsAD(allArrays,method="mds",color=as.character(comlist3),
        plotchar=pchlist3)
165 legend("bottomright",
166        c("fluid","blood TBPC","blood PTB","blood LTBI","blood healthy"),
167        pch=c(1,16,16,16,16),
168        col=c("red","red","orange","deepskyblue","green"),
169        cex=.8
170 )
171 export.plot(file.prefix=paste('fig',figure,'mds_all_IMPI_v3'),export.formats
        =export.formats.plots,height=height,width=(1 + sqrt(5))/2*height)
172
173 #3d PCA and scatterplot to check how blood and fluid separate in 3
        dimensions
174 plotSampleRelations3D_AD(allArrays,method="mds",color=as.character(comlist),
        backgr=as.character(comlist2))
175 rgl.snapshot( file.path(dir.figures,paste('fig',figure,'mds_all_IMPI_3D')),
        fmt="png", top=TRUE )
176
177 figure<-figure+1
178
179 #Sample relations of blood data
180
181 #colour factors for labeling plots by category
182 comlist.blood.diag<-as.factor(bloodArrays$class2)
183 levels(comlist.blood.diag)<-c("deepskyblue","orange","orange","green","red",
        "red","red")
184
185 comlist.blood.HIV<-as.factor(bloodArrays$HIV.Status)
186 levels(comlist.blood.HIV)<-c("green","red")
187 pchlist<-comlist.blood.HIV
188 levels(pchlist)<-c(21,24)
189 pchlist2<-as.numeric(as.vector(pchlist))
190 levels(pchlist)<-c(0,7)
191 pchlist3<-as.numeric(as.vector(pchlist))
192
193 plotSampleRelationsAD(bloodArrays,method="mds",color=as.character(comlist.
        blood.diag),plotchar=pchlist3)
194 legend("topright",
195        c("Healthy+","LTBI+","PTB+","TBPC+","Healthy-","LTBI-","PTB-","TBPC-"
        ),
196        pch=c(7,7,7,7,0,0,0,0),
197        col=c("green","deepskyblue","orange","red"),
```

```
        ncol=2,
        cex=.8
)
export.plot(file.prefix=paste('fig',figure,'mds_blood_dx_hiv'),export.formats=export.formats.plots,height=height,width=(1 + sqrt(5))/2*height)
figure<-figure+1


comlist.blood.sex<-as.factor(bloodArrays$SEX)
levels(comlist.blood.sex)<-c("red","deepskyblue")
comlist.blood.ster<-as.factor(bloodArrays$STEROIDS)
levels(comlist.blood.ster)<-c("grey","green","red")
pchlist<-comlist.blood.ster
levels(pchlist)<-c(0,6,17)
pchlist2<-as.numeric(as.vector(pchlist))

plotSampleRelationsAD(bloodArrays,method="mds",color=as.character(comlist.blood.sex),plotchar=pchlist2)
legend("topright",
        c("F?S","F-S","F+S","M?S","M-S","M+S"),
        pch=c(0,6,17,0,6,17),
        col=c("red","red","red","deepskyblue","deepskyblue","deepskyblue"),
        ncol=2,
        cex=.8
)

export.plot(file.prefix=paste('fig',figure,'mds_blood_sex_ster'),export.formats=export.formats.plots,height=height,width=(1 + sqrt(5))/2*height)
figure<-figure+1

#XP
comlist.blood.set<-bloodArrays$sample_name
pca<-grep("PCA",comlist.blood.set)
pc<-grep("PC[0-9]",comlist.blood.set)
eu<-grep("control",comlist.blood.set)

comlist.blood.set[eu]<-"blue"
comlist.blood.set[pca]<-"red"
comlist.blood.set[pc]<-"green"

plotSampleRelationsAD(bloodArrays,method="mds",color=as.character(comlist.blood.set),plotchar=15)
legend("topright",
        c("EU","SET2","SET1"),
        pch=c(15,15,15),
        col=c("blue","red","green"),
        ncol=2,
        cex=.8
)
export.plot(file.prefix=paste('fig',figure,'mds_blood_SETs'),export.formats=export.formats.plots,height=height,width=(1 + sqrt(5))/2*height)
figure<-figure+1
#XP


#Sample relations of fluid data

#colour factors for labeling plots by category
comlist.fluid.diag<-as.factor(fluidArrays$class2)
```

```
252 levels(comlist.fluid.diag)<-c("red","purple")
253
254 comlist.fluid.HIV<-as.factor(fluidArrays$HIV.Status)
255 levels(comlist.fluid.HIV)<-c("green","red")
256 pchlist<-comlist.fluid.HIV
257 levels(pchlist)<-c(21,24)
258 pchlist2<-as.numeric(as.vector(pchlist))
259 levels(pchlist)<-c(0,7)
260 pchlist3<-as.numeric(as.vector(pchlist))
261
262 plotSampleRelationsAD(fluidArrays,method="mds",color=as.character(comlist.
    fluid.diag),plotchar=pchlist3)
263 legend("bottomleft",
264        c("def+","prob+","def-","prob-"),
265        pch=c(7,7,0,0),
266        col=c("red","purple","red","purple"),
267        cex=.8
268 )
269 export.plot(file.prefix=paste('fig',figure,'mds_fluid_TBdx_hiv'),export.
    formats=export.formats.plots,height=height,width=(1 + sqrt(5))/2*height)
270 figure<-figure+1
```

Listing A.10: Overview of all data

A.7 Data manager for 7 questions

```
#metadata####
#IMPI_MA microarray analysis pipeline: question-based version
#Universal Data Manager
#Armin Deffur
#File created 10-06-2013
#Version 1.0

#System_setup.R####

source("/Users/armindeffur/Documents/002_Science_universal/Active_Research/
    IMPI-Microarray/Projects/01_scripts/06_IMPI-MA/Analysis/00_System/System
    _setup.R")

#Local Setup####

## Individual data folders (specific for each part)
dir.gx <- file.path(dir.data_root, '09_IMPI_MA_DATA/04_GX_format')
print(paste("GX12-format data repository", dir.gx))
dir.gs <- file.path(dir.data_root, '09_IMPI_MA_DATA_unversioned/05_
    GenomeStudio_reports')
print(paste("GenomeStudio report data repository", dir.gs))
dir.bgx <- file.path(dir.data_root, '09_IMPI_MA_DATA_unversioned/06_Illumina
     Manifest files/HT12v4')
print(paste("Illumina manifest file repository", dir.gx))
dir.jpg <- file.path(dir.data_root, '09_IMPI_MA_DATA_unversioned/07_
    imageData/JPG')
print(paste("Raw image data (.jpg) repository", dir.jpg))
dir.tif <- file.path(dir.data_root, '09_IMPI_MA_DATA_unversioned/07_
    imageData/TIFF')
print(paste("Raw image data (.tif) repository", dir.tif))
dir.pheno <- file.path(dir.data_root, '09_IMPI_MA_DATA/08_PhenoData')
print(paste("Phenodata data repository", dir.pheno))

#Output folders
dir.home<-dir.root
dir.main <- file.path(dir.root, '02_output/06_IMPI-MA','AllData')

#Current date string used for versioning output
ds<-Sys.time()

#versioned output
dir.output.version<-file.path(dir.main,paste("version_",ds))

## Define folder for storing results NB do this for each analysis
dir.results <- file.path(dir.output.version, "results")
print(paste("Results will be saved to", dir.results))

## Define folder for saving figures
dir.figures <- file.path(dir.output.version, "figures")
print(paste("Figures will be saved to", dir.figures))

## Define folder for saving RData files
dir.rdata <- file.path(dir.data_root,'09_IMPI_MA_DATA/06_RData_Q')
print(paste("Data will be saved to", dir.rdata))

for (dir in c(dir.main, dir.figures, dir.results)) {
```

```
51   if (!file.exists(dir)) {
52     dir.create(dir, recursive=T,showWarnings=T)
53   }
54 }
55
56 figure<-1
57
58 #WD####
59 setwd(dir.main)
60
61 #Data for Differential Expression
62
63 #Data for Deconvolution
64
65 #Data for WGCNA
66
67 #Import expression and phenotype data using lumi####
68
69 fileName<-file.path(dir.gx,"Armin_Hu_17_8_12_No uRNA_bkg sub_NO Norm__Sample
       _Probe_Profile.txt")
70 fileNameAnno<-file.path(dir.gs,"Armin_Hu_17_8_12_No uRNA_bkg sub_NO Norm_
       FinalReport.txt")
71 bgxfile<-file.path(dir.bgx,"HumanHT-12_V4_0_R1_15002873_B.bgx")
72
73 #PHENODATA ORIGINAL
74 #the phenodata below assigns TB status according to Pericardial culture.
       This misclassifies TB-PC suspects who were culture positive elsewhere as
        "probable" TB (wrong)
75 #phenodataPath<-file.path(dir.pheno,"phenoGS_edited_csv_colour_25_9_2012.csv
       ")
76
77 #PHENODATA EDIT 1
78 #the phenodata (class3 variable) below defines active TB as ANY positive
       site. This means that some probable TBPC cases are defnite TB cases.
       This is relevant as the !blood! signature will be determined by any (??)
        TB site. This needs to be tested where fluid is negative but another
       site positive. It is unclear if this applies to TBM, though. (issues of
       blood-brain-barrier, etc.)
79 #Important: when changing this, factors of diagnostic class may change,
       depending which "probable" case is made "definite".
80 #phenodataPath<-file.path(dir.pheno,"phenoGS_edited_csv_colour_28_10_2012.
       csv")
81
82 #PHENODATA EDIT 2
83 #With this in mind, the phenodata source file has been updated to change
       sample 20868control back to probableActiveTB, as the only positive
       culture was CSF. It seems unwise to include only one case of TBM, while
       there are many TB-PC and PTB.
84 #the phenodata file was edited: sample PCA141 blood and fluid assignments to
        chips have been exchanged, as earlier mds plots strongly suggested this
       . All results may change that depend on the correct assignment.
85
86 #IMPI-MA data
87 phenodataPath<-file.path(dir.pheno,"phenoGS_edited_csv_colour_17_01_2013.csv
       ")
88 sampleInfoDF<-read.csv(phenodataPath, header = TRUE, colClasses = "character
       ", comment.char = "",check.names = FALSE)
89 allArrays<-lumiR.batch(fileName,lib.mapping='lumiHumanIDMapping',
       sampleInfoFile=sampleInfoDF,verbose=TRUE)
```

```
colnames(exprs(allArrays))<-allArrays$sample_name

#Berry data
berry.sampleInfoDF.train<-read.csv("/Users/armindeffur/Documents/002_Science_universal/Active_Research/IMPI-Microarray/Data/03_TB_AOG/data/Berry_pheno/phenoData_train.csv", header = TRUE, colClasses = "character", comment.char = "",check.names = FALSE)
berry.train<-lumiR.batch("/Users/armindeffur/Documents/002_Science_universal/Active_Research/IMPI-Microarray/Data/03_TB_AOG/New_data/MatthewRequest_02Feb11_TrainingSet.txt",lib.mapping='lumiHumanIDMapping',sampleInfoFile=berry.sampleInfoDF.train,verbose=TRUE)

berry.sampleInfoDF.test<-read.csv("/Users/armindeffur/Documents/002_Science_universal/Active_Research/IMPI-Microarray/Data/03_TB_AOG/data/Berry_pheno/phenoData_test.csv", header = TRUE, colClasses = "character", comment.char = "",check.names = FALSE)
berry.test<-lumiR.batch("/Users/armindeffur/Documents/002_Science_universal/Active_Research/IMPI-Microarray/Data/03_TB_AOG/New_data/MatthewRequest_02Feb11_TestSet.txt",lib.mapping='lumiHumanIDMapping',sampleInfoFile=berry.sampleInfoDF.test,verbose=TRUE)

berry.sampleInfoDF.val<-read.csv("/Users/armindeffur/Documents/002_Science_universal/Active_Research/IMPI-Microarray/Data/03_TB_AOG/data/Berry_pheno/phenoData_val.csv", header = TRUE, colClasses = "character", comment.char = "",check.names = FALSE)
berry.val<-lumiR.batch("/Users/armindeffur/Documents/002_Science_universal/Active_Research/IMPI-Microarray/Data/03_TB_AOG/New_data/MatthewRequest_02Feb11_ValidationSet.txt",lib.mapping='lumiHumanIDMapping',sampleInfoFile=berry.sampleInfoDF.val,verbose=TRUE)

#subset allArrays by compartment
fluidArrays<-allArrays[,allArrays$Compartment=="fluid"]
bloodArrays<-allArrays[,allArrays$Compartment=="blood"]

#Data subsetting####

#CSC scheme: compartment, subset, contrast

save(berry.train,file=file.path(dir.rdata,'berry_train.RData'))
save(berry.test,file=file.path(dir.rdata,'berry_test.RData'))
save(berry.val,file=file.path(dir.rdata,'berry_val.RData'))

save(allArrays,file=file.path(dir.rdata,'IMPI_MA.RData'))

#Question 1: Tuberculosis
hivNegBloodArrays<-bloodArrays[,bloodArrays$HIV.Status=="negative"] #tb
B.Neg.AN<-hivNegBloodArrays[,hivNegBloodArrays$class3=="activeTB"|hivNegBloodArrays$class3=="notActiveTB"]#training
B.Neg.aN<-hivNegBloodArrays[,hivNegBloodArrays$class3=="probableActiveTB"|hivNegBloodArrays$class3=="notActiveTB"]#validation
save(B.Neg.AN,file=file.path(dir.rdata,'blood_hivNeg_aTB_noTB.RData'))
save(B.Neg.aN,file=file.path(dir.rdata,'blood_hivNeg_probTB_noTB.RData'))

hivPosBloodArrays<-bloodArrays[,bloodArrays$HIV.Status=="positive"]
B.Pos.AN<-hivPosBloodArrays[,hivPosBloodArrays$class3=="activeTB"|hivPosBloodArrays$class3=="notActiveTB"]
B.Pos.aN<-hivPosBloodArrays[,hivPosBloodArrays$class3=="probableActiveTB"|hivPosBloodArrays$class3=="notActiveTB"]
save(B.Pos.AN,file=file.path(dir.rdata,'blood_hivPos_aTB_noTB.RData'))
```

```
127 save(B.Pos.aN,file=file.path(dir.rdata,'blood_hivPos_probTB_noTB.RData'))
128
129 #Question 2: HIV in active TB
130 B.A.PosNeg<-bloodArrays[,bloodArrays$class3=="activeTB"] #tb
131 save(B.A.PosNeg,file=file.path(dir.rdata,'blood_aTB_HIVPosNeg.RData'))
132 PF.A.PosNeg<-fluidArrays[,fluidArrays$class3=="activeTB"]
133 save(PF.A.PosNeg,file=file.path(dir.rdata,'fluid_aTB_HIVPosNeg.RData'))
134 B.PTB.PosNeg<-bloodArrays[,bloodArrays$class2=="CON_PTB_definite"]
135 save(B.PTB.PosNeg,file=file.path(dir.rdata,'blood_defPTB_HIVPosNeg.RData'))
136 B.TBPC.PosNeg<-bloodArrays[,bloodArrays$class2=="TBPC_definite"]
137 save(B.TBPC.PosNeg,file=file.path(dir.rdata,'blood_defTBPC_HIVPosNeg.RData')
    )
138
139 #Question 3: Compartment
140 C.BPF<-allArrays[,allArrays$matching=="m"]
141 save(C.BPF,file=file.path(dir.rdata,'TBPC_matchedBloodFluid.RData'))
142 C.Neg.matchedBF<-C.BPF[,C.BPF$HIV.Status=="negative"]
143 save(C.Neg.matchedBF,file=file.path(dir.rdata,'TBPC_hivNeg_matchedBloodFluid
    .RData'))
144 C.Pos.matchedBF<-C.BPF[,C.BPF$HIV.Status=="positive"]
145 save(C.Pos.matchedBF,file=file.path(dir.rdata,'TBPC_hivPos_matchedBloodFluid
    .RData'))
146 C.Pos.EFF.matchedBF<-C.Pos.matchedBF[,C.Pos.matchedBF$ECP=="Eff"]
147 save(C.Pos.EFF.matchedBF,file=file.path(dir.rdata,'TBPC_hivPos_EFF_
    matchedBloodFluid.RData'))
148 C.Pos.EC.matchedBF<-C.Pos.matchedBF[,C.Pos.matchedBF$ECP=="EC"]
149 save(C.Pos.EC.matchedBF,file=file.path(dir.rdata,'TBPC_hivPos_EC_
    matchedBloodFluid.RData'))
150
151 #Question 4: LTBI
152 B.Neg.HL<-hivNegBloodArrays[,hivNegBloodArrays$class3=="notActiveTB"]
153 save(B.Neg.HL,file=file.path(dir.rdata,'blood_hivNeg_H_L.RData'))
154 B.Pos.HL<-hivPosBloodArrays[,hivPosBloodArrays$class3=="notActiveTB"]
155 save(B.Pos.HL,file=file.path(dir.rdata,'blood_hivPos_H_L.RData'))
156
157 #Question 5: HIV in not active TB
158 B.N.PosNeg<-bloodArrays[,bloodArrays$class3=="notActiveTB"]
159 save(B.N.PosNeg,file=file.path(dir.rdata,'blood_notActiveTB_PosNeg.RData'))
160 B.H.PosNeg<-bloodArrays[,bloodArrays$class2=="CON_healthy"]
161 save(B.H.PosNeg,file=file.path(dir.rdata,'blood_healthy_PosNeg.RData'))
162 B.L.PosNeg<-bloodArrays[,bloodArrays$class2=="CON_LTBI"]
163 save(B.L.PosNeg,file=file.path(dir.rdata,'blood_LTBI_PosNeg.RData'))
164
165 #Question 6: TB site
166 B.Pos.PTB_TBPC<-hivPosBloodArrays[,hivPosBloodArrays$class3=="activeTB"]
167 save(B.Pos.PTB_TBPC,file=file.path(dir.rdata,'blood_hivPos_PTB_TBPC.RData'))
168 B.Neg.PTB_TBPC<-hivNegBloodArrays[,hivNegBloodArrays$class3=="activeTB"]
169 save(B.Neg.PTB_TBPC,file=file.path(dir.rdata,'blood_hivNeg_PTB_TBPC.RData'))
170
171 #Question 7: Haemodynamic phenotype
172 B.C.Pos.EffEC<-hivPosBloodArrays[,hivPosBloodArrays$ECP=="Eff"|
    hivPosBloodArrays$ECP=="EC"]
173 save(B.C.Pos.EffEC,file=file.path(dir.rdata,'blood_TBPC_hivPos_EffEC.RData')
    )
174 PF.C.Pos.EffEC<-fluidArrays[,(fluidArrays$ECP=="Eff"|fluidArrays$ECP=="EC")&
    fluidArrays$HIV.Status=="positive"]
175 save(PF.C.Pos.EffEC,file=file.path(dir.rdata,'fluid_TBPC_hivPos_EffEC.RData'
    ))
176
```

```
177 #Signatures
178
179 #Berry signatures
180 #file=file.path(dir.gx,"Nature_paper_transcripts","P22_42_04Mar09_Training_
        393_GX11.txt")
181 #np393<-read.table(file,header=TRUE,sep="\t")
182 #char393<-as.character(np393[,1])
183
184 #berry.sig<-char393
185 #save(berry.sig,file=file.path(dir.rdata,'berry.sig.RData'))
186
187 #need to get 86 transcript list; downloaded as xls
188
189 #import
190 file86=file.path(dir.gx,"Nature_paper_transcripts","nature09247-s4.csv")
191 np86<-read.csv(file86,header=TRUE,sep=",")
192 berry.86<-np86
193 save(berry.86,file=file.path(dir.rdata,'berry.86.RData'))
194
195 #import manual berry393
196
197 #not done, need to fix format...
198 file393=file.path(dir.gx,"Nature_paper_transcripts","nature09247-s2.csv")
199 np393<-read.csv(file393,header=TRUE,sep=",")
200 berry.393<-np393
201 save(berry.393,file=file.path(dir.rdata,'berry.393.RData'))
```

Listing A.11: Data-manager-Q.R

A.8 Analysis: IMPI-MA

```
##TO DO
#1. table 1 code

#setup####
#Current date string used for versioning output

source("/Users/armindeffur/Documents/002_Science_universal/Active_Research/IMPI-Microarray/Projects/01_scripts/06_IMPI-MA/Analysis/00_System/System_setup.R")
source(file.path(dir.R.files,"superHeatmap.R"))
source(file.path(dir.R.files,"superHeatmap2.R"))
par("xpd"=FALSE)
## Individual data folders (specific for each part)
dir.rdata <- file.path(dir.data_root,'09_IMPI_MA_DATA/06_RData_Q')

#Output folders
dir.home<-dir.root
dir.main <- file.path(dir.root, '02_output/06_IMPI-MA/DE_DC_CN','DE_AD_TB')
ds<-Sys.time()
#versioned output
dir.output.version<-file.path(dir.main,paste("version_",ds))

## Define folder for storing results NB do this for each analysis
dir.results <- file.path(dir.output.version, "results")

## Define folder for saving figures
dir.figures <- file.path(dir.output.version, "figures")

for (dir in c(dir.main, dir.figures, dir.results)) {
  if (!file.exists(dir)) {
    dir.create(dir, recursive=T,showWarnings=T)
  }
}

#sink(type="output",file=file.path(dir.results,"session_output.txt"))
sessionInfo()
print(paste("Begin time:",ds))
print(paste("Results will be saved to", dir.results))
print(paste("Figures will be saved to", dir.figures))

#all iterators and their definitions
#q.i: loop over questions (n=7)
#filename: loop over filenames when loading RDtata files
#dataindex: loop over datasets when making datalist, colourmaps

#Make questions
#Questions constructs####

questions=list(
  "Tuberculosis"=list(
    "datalist_filenames"=list("blood_hivNeg_aTB_noTB.RData","blood_hivPos_aTB_noTB.RData"),
    "dataset_variables"=list("B.Neg.AN","B.Pos.AN"),
    "dataset_names"=list("Blood HIVneg activeTB-notActiveTB","Blood HIVpos activeTB-notActiveTB"),
    "contrast_variable"="class3",
```

```
    "colour_variable"="class",
    "colourmap"=list("deepskyblue"="CON_LTBI","orange"="CON_PTB","green"="CON_healthy","red"="TBPC"),
    "pqlist"=list(illuminaHumanv4PROBEQUALITY,illuminaHumanv4PROBEQUALITY),
    "wgcna_pheno"=c(),#column numbers in phenodata specific for the question; may be empty
    "wgcna_contrast"=list()
  ),
  "TBSite"=list(
    "datalist_filenames"=list("blood_hivPos_PTB_TBPC.RData","blood_hivNeg_PTB_TBPC.RData"),
    "dataset_variables"=list("B.Pos.PTB_TBPC","B.Neg.PTB_TBPC"),
    "dataset_names"=list("Blood HIVpos PTB-TBPC","Blood HIVneg PTB-TBPC"),
    "contrast_variable"="class",
    "colour_variable"="class",
    "colourmap"=list("orange"="CON_PTB","red"="TBPC"),
    "pqlist"=list(illuminaHumanv4PROBEQUALITY,illuminaHumanv4PROBEQUALITY),
    "wgcna_pheno"=c()#column numbers in phenodata specific for the question; may be empty
  ),
  "LTBI"=list(
    "datalist_filenames"=list("blood_hivPos_H_L.RData","blood_hivNeg_H_L.RData"),
    "dataset_variables"=list("B.Pos.HL","B.Neg.HL"),
    "dataset_names"=list("Blood HIVpos Healthy-LTBI","Blood HIVneg Healthy-LTBI"),
    "contrast_variable"="class",
    "colour_variable"="class",
    "colourmap"=list("green"="CON_healthy","deepskyblue"="CON_LTBI"),
    "pqlist"=list(illuminaHumanv4PROBEQUALITY,illuminaHumanv4PROBEQUALITY),
    "wgcna_pheno"=c()#column numbers in phenodata specific for the question; may be empty
  ),
  "hdPhenotype"=list(
    "datalist_filenames"=list("blood_TBPC_hivPos_EffEC.RData","fluid_TBPC_hivPos_EffEC.RData"),
    "dataset_variables"=list("B.C.Pos.EffEC","PF.C.Pos.EffEC"),
    "dataset_names"=list("Blood TBPC HIVPos Eff-EC","Fluid TBPC HIVPos Eff-EC"),
    "contrast_variable"="ECP",
    "colour_variable"="ECP",
    "colourmap"=list("blue"="Eff","yellow"="EC"),
    "pqlist"=list(illuminaHumanv4PROBEQUALITY,illuminaHumanv4PROBEQUALITY),
    "wgcna_pheno"=c()#column numbers in phenodata specific for the question; may be empty
  ),
  "hiv_noTB"=list(
    "datalist_filenames"=list("blood_notActiveTB_PosNeg.RData","blood_LTBI_PosNeg.RData","blood_healthy_PosNeg.RData"),
    "dataset_variables"=list("B.N.PosNeg","B.L.PosNeg","B.H.PosNeg"),
    "dataset_names"=list("Blood not activeTB HIVpos-HIVneg","Blood LTBI HIVpos-HIVneg","Blood Healthy HIVpos-HIVneg"),
    "contrast_variable"="HIV.Status",
    "colour_variable"="HIV.Status",
    "colourmap"=list("green"="negative","red"="positive"),
    "pqlist"=list(illuminaHumanv4PROBEQUALITY,illuminaHumanv4PROBEQUALITY,illuminaHumanv4PROBEQUALITY),
    "wgcna_pheno"=c()#column numbers in phenodata specific for the question; may be empty
```

```
  ),
  "hiv_TB"=list(
    "datalist_filenames"=list("blood_aTB_HIVPosNeg.RData","fluid_aTB_HIVPosNeg.RData","blood_defPTB_HIVPosNeg.RData","blood_defTBPC_HIVPosNeg.RData"),
    "dataset_variables"=list("B.A.PosNeg","PF.A.PosNeg","B.PTB.PosNeg","B.TBPC.PosNeg"),
    "dataset_names"=list("Blood activeTB HIVpos-HIVneg","Fluid activeTB HIVpos-HIVneg","Blood PTB HIVpos-HIVneg","Blood TB-PC HIVpos-HIVneg"),
    "contrast_variable"="HIV.Status",
    "colour_variable"="HIV.Status",
    "colourmap"=list("green"="negative","red"="positive"),
    "pqlist"=list(illuminaHumanv4PROBEQUALITY,illuminaHumanv4PROBEQUALITY,illuminaHumanv4PROBEQUALITY,illuminaHumanv4PROBEQUALITY),
    "wgcna_pheno"=c()#column numbers in phenodata specific for the question; may be empty
  ),
  "Compartment"=list(
    "datalist_filenames"=list("TBPC_matchedBloodFluid.RData","TBPC_hivNeg_matchedBloodFluid.RData","TBPC_hivPos_matchedBloodFluid.RData","TBPC_hivPos_EFF_matchedBloodFluid.RData","TBPC_hivPos_EC_matchedBloodFluid.RData"),
    "dataset_variables"=list("C.BPF","C.Neg.matchedBF","C.Pos.matchedBF","C.Pos.EFF.matchedBF","C.Pos.EC.matchedBF"),
    "dataset_names"=list("TB-PC all blood-fluid","TB-PC HIVneg blood-fluid","TB-PC HIVpos blood-fluid","TB-PC HIVpos EFF blood-fluid","TB-PC HIVpos EC blood-fluid"),
    "contrast_variable"="Compartment",
    "colour_variable"="Compartment",
    "colourmap"=list("red"="blood","yellow"="fluid"),
    "pqlist"=list(illuminaHumanv4PROBEQUALITY,illuminaHumanv4PROBEQUALITY,illuminaHumanv4PROBEQUALITY,illuminaHumanv4PROBEQUALITY,illuminaHumanv4PROBEQUALITY),
    "wgcna_pheno"=c()#column numbers in phenodata specific for the question; may be empty
  )
)

#for loops####
#q.i is question iterator; length = 7

#debug
#q.i<-1

#real
for (q.i in 1:length(questions)){
#Question (c.q: "current question")
c.q=questions[[q.i]]

#definitions for the current question####
datalist.filenames=c.q$datalist_filenames
dataset.variables=c.q$dataset_variables
dataset.names=c.q$dataset_names
contrast.variable=c.q$contrast_variable
colour.variable=c.q$colour_variable
colourmap=c.q$colourmap
pqlist=c.q$pqlist
extra.traits<-c.q$wgcna_pheno

```

```
142 print(paste("Results for Question ",q.i,": ",names(questions)[q.i],sep=""))
143
144 #load data and make datalist####
145
146 for (filename in datalist.filenames){
147 load(file.path(dir.rdata,filename))
148 }
149
150 datalist<-list()
151 for (dataindex in 1:length(datalist.filenames)){
152 datalist[[dataindex]]<-eval(parse(text=dataset.variables[[dataindex]]))
153 names(datalist)[[dataindex]]<-dataset.names[[dataindex]]
154 }
155
156 #make pqlist####
157 pqlist<-c.q$pqlist
158
159 print(paste(c("Data to be analysed:",names(datalist))))
160
161 #generate table 1####
162 print("Table 1 data generation")
163 #
164 #Table 1: Compare groups across datasets
165
166 #contrasts
167 contr<-lapply(dataset.variables,function(any){
168   as.factor(eval(parse(text=paste(any,"$",contrast.variable,sep=""))))
169 }
170   )
171 #AGE####
172 #####AGE across all
173 ageAA<-lapply(datalist,function(part){as.numeric(pData(part)$AGE_CALC)})
174 n.age<-lapply(ageAA,length)
175 sum.age<-lapply(ageAA,summary)
176 test.age<-try(kruskal.test(ageAA))
177 if(class(test.age)=="try-error"){
178   test.age$p.value=NA
179   test.age$method="NA"}
180
181 #####AGE across (contrast1)
182 ageAC1<-lapply(datalist,function(part){
183   as.numeric(pData(part)
184              [eval(parse(text=paste("pData(part)$",contrast.variable,sep="")
     ))
185               ==levels(
186                as.factor(eval(parse(text=paste(dataset.variables,"$",
     contrast.variable,sep=""))))[[1]],]
187              $AGE_CALC)
188   }
189                )
190
191 n.age.C1<-lapply(ageAC1,length)
192 sum.age.C1<-lapply(ageAC1,summary)
193 test.age.C1<-try(kruskal.test(ageAC1))
194
195 if(class(test.age.C1)=="try-error"){
196   test.age.C1$p.value=NA
197   test.age.C1$method="NA"}
198
```

```
199
200 #####AGE across (contrast1)
201 ageAC2<-lapply(datalist,function(part){
202   as.numeric(pData(part)
203              [eval(parse(text=paste("pData(part)$",contrast.variable,sep="")
    ))
204               ==levels(
205                 as.factor(eval(parse(text=paste(dataset.variables,"$",
    contrast.variable,sep=""))))[[2]],]
206              $AGE_CALC)
207 }
208 )
209
210 n.age.C2<-lapply(ageAC2,length)
211 sum.age.C2<-lapply(ageAC2,summary)
212 test.age.C2<-try(kruskal.test(ageAC2))
213
214 if(class(test.age.C2)=="try-error"){
215   test.age.C2$p.value=NA
216   test.age.C2$method="NA"}
217
218
219
220 #####AGE down (all sets)
221
222 age.down.data<-list()
223 age.down.n<-list()
224 age.down.sum<-list()
225 age.down.test<-list()
226 for (set in 1:length(datalist)){
227 age.down.data[[set]]<-list(ageAC1[[set]],ageAC2[[set]])
228 age.down.n[[set]]<-lapply(age.down.data[[set]],length)
229 age.down.sum[[set]]<-lapply(age.down.data[[set]],summary)
230 age.down.test[[set]]<-try(kruskal.test(age.down.data[[set]]))
231 if(class(age.down.test[[set]])=="try-error"){
232   age.down.test[[set]]$p.value=NA
233   age.down.test[[set]]$method="NA"}
234
235 }
236 #SEX####
237 #SEX all across
238 sexAA<-lapply(datalist,function(part){as.factor(pData(part)$SEX)})
239 for(sexindex in 1:length(sexAA)){levels(sexAA[[sexindex]])<-c("Female","Male
    ")}
240
241 sexAAdata<-t(matrix(as.numeric(unlist(lapply(sexAA,function(z){rbind(as.
    vector(table(z)))}))),ncol=length(sexAA),dimnames=list(c("Female","Male"
    ),unlist(dataset.names))))
242 sexAA.test<-try(if(length(sexAA)==2){prop.test(sexAAdata)}else{fisher.test(
    sexAAdata)})
243 if(class(sexAA.test)=="try-error"){
244   sexAA.test$p.value=NA
245   sexAA.test$method="NA"}
246
247 #####Sex across (contrast1)
248 sexAC1<-lapply(datalist,function(part){
249   as.factor(pData(part)
250              [eval(parse(text=paste("pData(part)$",contrast.variable,sep="")
    ))
```

```
251                ==levels(
252                  as.factor(eval(parse(text=paste(dataset.variables,"$",
    contrast.variable,sep="")))))[[1]],]
253              $SEX)
254 }
255 )
256
257 for(sexindex in 1:length(sexAC1)){levels(sexAC1[[sexindex]])<-c("Female","
    Male")}
258 sexAC1data<-t(matrix(as.numeric(unlist(lapply(sexAC1,function(z){rbind(as.
    vector(table(z)))}))),ncol=length(sexAC1),dimnames=list(c("Female","Male
    "),unlist(dataset.names))))
259 sexAC1.test<-try(if(length(sexAC1)==2){prop.test(sexAC1data)}else{fisher.
    test(sexAC1data)})
260
261 if(class(sexAC1.test)=="try-error"){
262   sexAC1.test$p.value=NA
263   sexAC1.test$method="NA"}
264
265
266 #####Sex across (contrast2)
267 sexAC2<-lapply(datalist,function(part){
268   as.factor(pData(part)
269             [eval(parse(text=paste("pData(part)$",contrast.variable,sep="")))
    ]
270              ==levels(
271                as.factor(eval(parse(text=paste(dataset.variables,"$",
    contrast.variable,sep="")))))[[2]],]
272             $SEX)
273 }
274 )
275
276 for(sexindex in 1:length(sexAC2)){levels(sexAC2[[sexindex]])<-c("Female","
    Male")}
277
278 sexAC2data<-t(matrix(as.numeric(unlist(lapply(sexAC2,function(z){rbind(as.
    vector(table(z)))}))),ncol=length(sexAC2),dimnames=list(c("Female","Male
    "),unlist(dataset.names))))
279 sexAC2.test<-try(if(length(sexAC2)==2){prop.test(sexAC2data)}else{fisher.
    test(sexAC2data)})
280
281 if(class(sexAC2.test)=="try-error"){
282   sexAC2.test$p.value=NA
283   sexAC2.test$method="NA"}
284
285 #####Sex down
286 sex.down.data<-list()
287 sex.down.test<-list()
288 for (set in 1:length(datalist)){
289   sex.down.data[[set]]<-matrix(rbind(sexAC1data[set,],sexAC2data[set,]),ncol
      =2,dimnames=list(levels(contr[[1]]),c("Female","Male")))
290   sex.down.test[[set]]<-try(prop.test(sex.down.data[[set]]))
291   if(class(sex.down.test[[set]])=="try-error"){
292     sex.down.test[[set]]$p.value=NA
293     sex.down.test[[set]]$method="NA"}
294 }
295
296
297 #CD4
```

```
#CD4####
#####CD4 across all
CD4AA<-lapply(datalist,function(part){as.numeric(pData(part)$CD4)})
n.CD4<-lapply(CD4AA,length)
sum.CD4<-lapply(CD4AA,summary)
test.CD4<-try(kruskal.test(CD4AA))

if(class(test.CD4)=="try-error"){
  test.CD4$p.value=NA
  test.CD4$method="NA"}

#####CD4 across (contrast1)
CD4AC1<-lapply(datalist,function(part){
  as.numeric(pData(part)
             [eval(parse(text=paste("pData(part)$",contrast.variable,sep="")
    ))
              ==levels(
                as.factor(eval(parse(text=paste(dataset.variables,"$",
    contrast.variable,sep="")))))[[1]],]
             $CD4)
}
)

n.CD4.C1<-lapply(CD4AC1,length)
sum.CD4.C1<-lapply(CD4AC1,summary)
test.CD4.C1<-try(kruskal.test(CD4AC1))

if(class(test.CD4.C1)=="try-error"){
  test.CD4.C1$p.value=NA
  test.CD4.C1$method="NA"}

#####CD4 across (contrast2)
CD4AC2<-lapply(datalist,function(part){
  as.numeric(pData(part)
             [eval(parse(text=paste("pData(part)$",contrast.variable,sep="")
    ))
              ==levels(
                as.factor(eval(parse(text=paste(dataset.variables,"$",
    contrast.variable,sep="")))))[[2]],]
             $CD4)
}
)

n.CD4.C2<-lapply(CD4AC2,length)
sum.CD4.C2<-lapply(CD4AC2,summary)
test.CD4.C2<-try(kruskal.test(CD4AC2))

if(class(test.CD4.C2)=="try-error"){
  test.CD4.C2$p.value=NA
  test.CD4.C2$method="NA"}
#####CD4 down (all sets)

CD4.down.data<-list()
CD4.down.n<-list()
CD4.down.sum<-list()
CD4.down.test<-list()
for (set in 1:length(datalist)){
  CD4.down.data[[set]]<-list(CD4AC1[[set]],CD4AC2[[set]])
  CD4.down.n[[set]]<-lapply(CD4.down.data[[set]],length)
```

```
  CD4.down.sum[[set]]<-lapply(CD4.down.data[[set]],summary)
  CD4.down.test[[set]]<-try(kruskal.test(CD4.down.data[[set]]))
  if(class(CD4.down.test[[set]])=="try-error"){
    CD4.down.test[[set]]$p.value=NA
    CD4.down.test[[set]]$method="NA"}
}
#print and export table 1 for each question####

names.across<-c(unlist(lapply(dataset.variables,function(x){c(paste(x,"Med")
    ,paste(x,"LQ"),paste(x,"UQ"))})),"P val","Test")
age.line0<-c(rep(c("Median","LQ","UQ"),length(dataset.names)),"Pval","Test")
age.line1<-c(unlist(lapply(n.age,function(x){c(x,"","")})),"","")
age.line2<-unlist(list(lapply(sum.age,function(x){x[c(3,2,5)]}),list(sprintf
    ("%.3f",test.age$p.value)),list(substr(test.age$method,1,20))))
age.line3<-unlist(list(lapply(sum.age.C1,function(x){x[c(3,2,5)]}),list(
    sprintf("%.3f",test.age.C1$p.value)),list(substr(test.age.C1$method
    ,1,20))))
age.line4<-unlist(list(lapply(sum.age.C2,function(x){x[c(3,2,5)]}),list(
    sprintf("%.3f",test.age.C2$p.value)),list(substr(test.age.C2$method
    ,1,20))))
age.line5<-unlist(list(unlist(lapply(age.down.test,function(x){c(sprintf("
    %.3f",x$p.value),"","")})),list("","")),recursive=T)
age.line6<-unlist(list(unlist(lapply(age.down.test,function(x){c(substr(x$
    method,1,20),"","")})),list("","")),recursive=T)

sex.line0<-c(rep(c("prop F","Female","Male"),length(dataset.names)),"Pval","
    test")
sex.line1<-c(as.character(apply(sexAAdata,1,function(x){
  c(
  sprintf("%.3f",x[1]/x[2]),
  x[1],
  x[2]
    )
  })),sprintf("%.3f",sexAA.test$p.value),
                    substr(sexAA.test$method,1,20))
sex.line2<-c(as.character(apply(sexAC1data,1,function(x){
  c(
    sprintf("%.3f",x[1]/x[2]),
    x[1],
    x[2]
     )
})),sprintf("%.3f",sexAC1.test$p.value),
              substr(sexAC1.test$method,1,20))

sex.line3<-c(as.character(apply(sexAC2data,1,function(x){
  c(
    sprintf("%.3f",x[1]/x[2]),
    x[1],
    x[2]
     )
})),sprintf("%.3f",sexAC2.test$p.value),
              substr(sexAC2.test$method,1,20))
sex.line4<-unlist(list(unlist(lapply(sex.down.test,function(x){c(sprintf("
    %.3f",x$p.value),"","")})),list("","")),recursive=T)
sex.line5<-unlist(list(unlist(lapply(sex.down.test,function(x){c(substr(x$
    method,1,20),"","")})),list("","")),recursive=T)

CD4.line0<-c(rep(c("Median","LQ","UQ"),length(dataset.names)),"Pval","Test")
CD4.line1<-c(unlist(lapply(n.CD4,function(x){c(x,"","")})),"","")
```

```
401 CD4.line2<-unlist(list(lapply(sum.CD4,function(x){x[c(3,2,5)]}),list(sprintf
      ("%.3f",test.CD4$p.value)),list(substr(test.CD4$method,1,20))))
402 CD4.line3<-unlist(list(lapply(sum.CD4.C1,function(x){x[c(3,2,5)]}),list(
      sprintf("%.3f",test.CD4.C1$p.value)),list(substr(test.CD4.C1$method
      ,1,20))))
403 CD4.line4<-unlist(list(lapply(sum.CD4.C2,function(x){x[c(3,2,5)]}),list(
      sprintf("%.3f",test.CD4.C2$p.value)),list(substr(test.CD4.C2$method
      ,1,20))))
404 CD4.line5<-unlist(list(unlist(lapply(CD4.down.test,function(x){c(sprintf("
      %.3f",x$p.value),"","")})),list("","")),recursive=T)
405 CD4.line6<-unlist(list(unlist(lapply(CD4.down.test,function(x){c(substr(x$
      method,1,20),"","")})),list("","")),recursive=T)
406
407
408 dft<-data.frame(rbind(
409   age.line0,age.line2,age.line3,age.line4,age.line5,age.line6,
410   sex.line0,sex.line1,sex.line2,sex.line3,sex.line4,sex.line5,
411   CD4.line0,CD4.line2,CD4.line3,CD4.line4,CD4.line5,CD4.line6
412 ))
413
414 names.down<-c("Age","age_All",paste("age_",levels(contr[[1]])),"age_P value"
      ,"age_Test",
415               "Sex","sex_All",paste("sex_",levels(contr[[1]])),"sex_P value"
      ,"sex_Test",
416               "CD4","CD4_All",paste("CD4_",levels(contr[[1]])),"CD4_P value"
      ,"CD4_Test"
417 )
418
419 row.names(dft)<-c(names.down)
420
421 names(dft)<-unlist(c(lapply(dataset.names,function(x){c(x,"","")}),"Pval","
      Test"))
422
423 #hardcoded!
424 #xt<-xtable(dft,align=c("p{1cm}|","p{1cm}|","p{0.8cm}|","p{0.8cm}|","p{1cm
      }|","p{0.8cm}|","p{0.8cm}|","p{0.8cm}|","p{1cm}"))
425 xt<-xtable(dft,align=c("|",rep(c("p{1cm}|"),ncol(dft)),"p{1cm}|"))
426
427 print(xt,hline.after=-1:nrow(dft),include.colnames=T,size=c("tiny"),file=(
428   file.path(dir.results,paste("Table 1 for question ",q.i,": ",names(
        questions)[[q.i]],".tex",sep=""))
429   ))
430
431
432
433 #temporary q.i closure
434 #}
435
436 #start analysis####
437
438 #analysis for loop
439 for (i in 1:length(datalist)){
440
441 print(paste("current dataset: ",names(datalist)[[i]]))
442 ##DE analysis####
443 analysis="DE"
444 an.count<-1
445 print("Analysis 1: Differential expression")
446
```

```
447 #define empty lists required for DE analysis####
448 tbstat=list()
449
450 datalist.v<-list()
451 datalist.q<-list()
452 data.fil<-list()
453 data.v.fil<-list()
454
455 tbstat.class=list()
456 classcolours=list()
457
458 design<-list()
459 contrast<-list()
460 fit<-list()
461 fit2<-list()
462 gr<-list()
463 vc<-list()
464 gr2<-list()
465 vc<-list()
466 vc2<-list()
467 reslist<-list()
468 reslistTT<-list()
469 reslistTTall<-list()
470 reslistTT30<-list()
471 reslistTT100<-list()
472 reslistTT2000<-list()
473 reslistTT8000<-list()
474 reslistTT16000<-list()
475
476 #Make class levels and colours####
477
478 #tbstat=list(as.factor(B.N.AX$class2),as.factor(B.P.AX$class2))
479 tbstat[[i]]<-as.factor(eval(parse(text=paste(dataset.variables[[i]],"$",
        colour.variable,sep=""))))
480 #levels(tbstat[[i]])<-colourmap
481 #}
482
483 #for (dataindex in 1:length(datalist)){
484 classcolours[[i]]<-tbstat[[i]]
485 levels(classcolours[[i]])<-colourmap
486 #}
487
488 #Show raw data####
489
490 #boxplots and stripcharts
491 figure<-1
492 setwd(dir.figures)
493
494 #fix this by integrating into for loop
495 colnames(exprs(datalist[[i]]))<-eval(parse(text=paste(dataset.variables[[i
        ]],"$","sample_name",sep="")))
496 #colnames(exprs(datalist[[1]]))<-B.N.AX$sample_name
497 #colnames(exprs(datalist[[2]]))<-B.P.AX$sample_name
498
499
500 # the order of this depends completely on the order in the csv file - it
        should not impact on the actual analysis. Just need to make sure that
        the class assignments for DE are correct
501
```

```
#for (i in 1:length(datalist)){
  boxplot(datalist[[i]],main=names(datalist)[i],col=as.character(
    classcolours[[i]]))
  export.plot(file.prefix=paste('fig',q.i,'.',an.count,'.',i,'.',figure,
    names(questions)[[q.i]],'.',analysis,names(datalist)[i],'boxplot'),
    export.formats=export.formats.plots,height=height,width=(1 + sqrt(5))/2*
    height)
#}

figure<-figure+1

#Outlier plot
#for (i in 1:length(datalist)){
  plot(datalist[[i]],what="outlier",main=names(datalist)[i])
  export.plot(file.prefix=paste('fig',q.i,'.',an.count,'.',i,'.',figure,
    names(questions)[[q.i]],'.',analysis,names(datalist)[i],'outlier_plot'),
    export.formats=export.formats.plots,height=height,width=(1 + sqrt(5))/2*
    height)
#}
figure<-figure+1

#MDS all probes

#for (i in 1:length(datalist)){
  plotSampleRelationsAD(datalist[[i]],method="mds",color=as.character(
    classcolours[[i]]),plotchar=16)
  legend("bottomleft",
         levels(tbstat[[i]]),
         pch=16,
         col=levels(classcolours[[i]]),
         cex=.8
  )
  export.plot(file.prefix=paste('fig',q.i,'.',an.count,'.',i,'.',figure,
    names(questions)[[q.i]],'.',analysis,names(datalist)[i],"MDS_all"),
    export.formats=export.formats.plots,height=height,width=(1 + sqrt(5))/2*
    height)
#}
figure<-figure+1

#PCA all probes
#for (i in 1:length(datalist)){
  l=colnames(exprs(datalist[[i]]))
  #trueClasses <-tbstat
  spca <- SamplePCA(exprs(datalist[[i]]), tbstat[[i]])
  plot(spca, col=levels(classcolours[[i]]),main=paste(nrow(exprs(datalist[[i
    ]])),"probes"))#,cex=2,cex.axis=2,cex.main=3,cex.lab=2.5,pch=16)
  mtext(sprintf("%.2f",spca@variances[1]/sum(spca@variances)),side=1,line=3,
    adj=1,cex=1.5)
  mtext(sprintf("%.2f",spca@variances[2]/sum(spca@variances)),side=2,line=3,
    adj=1,cex=1.5)
  legend("bottomleft",
         levels(tbstat[[i]]),
         pch=16,
         col=levels(classcolours[[i]]),
         cex=.8
  )
  # mark the group centers

for (type in 1:length(levels(tbstat[[i]]))){
```

```
547   x1 <- predict(spca, matrix(apply(t(exprs(datalist[[i]])[,grep(levels(
        tbstat[[i]])[[type]],eval(parse(text=paste("datalist[[i]]$",colour.
        variable,sep=""))))]), 2, mean), ncol=1))
548   points(x1[1], x1[2], col=levels(classcolours[[i]])[[type]], cex=6,pch=9)}
549
550
551   export.plot(file.prefix=paste('fig',q.i,'.',an.count,'.',i,'.',figure,
        names(questions)[[q.i]],'.',analysis,names(datalist)[i],"PCA_all"),
        export.formats=export.formats.plots,height=height,width=(1 + sqrt(5))/2*
        height)
552 #}
553 figure<-figure+1
554
555 #preprocess####
556
557 #Variance stabilising transformation
558 #datalist.v<-list()
559 #for (i in 1:length(datalist)){
560   datalist.v[[i]]<-lumiT(datalist[[i]],simpleOutput=FALSE)
561 #}
562 #Quantile normalise
563 #datalist.q<-list()
564 #for (i in 1:length(datalist.v)){
565   datalist.q[[i]]<-lumiN(datalist.v[[i]],method="quantile")
566 #}
567
568 #show preprocessed data####
569 #boxplots of quantile normalised data
570 #for (i in 1:length(datalist.q)){
571   boxplot(datalist.q[[i]],main=names(datalist.q)[i],col=as.character(
        classcolours[[i]]))
572   export.plot(file.prefix=paste('fig',q.i,'.',an.count,'.',i,'.',figure,
        names(questions)[[q.i]],'.',analysis,names(datalist)[i],'boxplot (q norm
        )'),export.formats=export.formats.plots,height=height,width=(1 + sqrt(5)
        )/2*height)
573 #}
574 figure<-figure+1
575
576 #density plots of quantile normalised data
577 #for (i in 1:length(datalist.q)){
578   density(datalist.q[[i]],main=names(datalist)[i],col=as.character(
        classcolours[[i]]),addLegend=FALSE)
579   export.plot(file.prefix=paste('fig',q.i,'.',an.count,'.',i,'.',figure,
        names(questions)[[q.i]],'.',analysis,names(datalist)[i],'density plot
        qnorm'),export.formats=export.formats.plots,height=height,width=(1 +
        sqrt(5))/2*height)
580 #}
581 figure<-figure+1
582
583 #MDS all probes (normalised)
584
585 #for (i in 1:length(datalist.q)){
586   plotSampleRelationsAD(datalist.q[[i]],method="mds",color=as.character(
        classcolours[[i]]),plotchar=16)
587 legend("bottomleft",
588        levels(tbstat[[i]]),
589        pch=16,
590        col=levels(classcolours[[i]]),
591        cex=.8)
```

```
592   export.plot(file.prefix=paste('fig',q.i,'.',an.count,'.',i,'.',figure,
        names(questions)[[q.i]],'.',analysis,names(datalist)[i],"MDS_norm"),
        export.formats=export.formats.plots,height=height,width=(1 + sqrt(5))/2*
        height)
593 #}
594 figure<-figure+1
595
596
597 #PCA all probes (normalised)
598 #for (i in 1:length(datalist.q)){
599   l=colnames(exprs(datalist.q[[i]]))
600   spca <- SamplePCA(exprs(datalist.q[[i]]), tbstat[[i]])
601   plot(spca, col=levels(classcolours[[i]]),main=paste(nrow(exprs(datalist.q
        [[i]])),"probes (normalised)"))#,cex=2,cex.axis=2,cex.main=3,cex.lab
        =2.5,pch=16)
602   mtext(sprintf("%.2f",spca@variances[1]/sum(spca@variances)),side=1,line=3,
        adj=1,cex=1.5)
603   mtext(sprintf("%.2f",spca@variances[2]/sum(spca@variances)),side=2,line=3,
        adj=1,cex=1.5)
604 legend("bottomleft",
605        levels(tbstat[[i]]),
606        pch=16,
607        col=levels(classcolours[[i]]),
608        cex=.8)
609   # mark the group centers
610 for (type in 1:length(levels(tbstat[[i]]))){
611   x1 <- predict(spca, matrix(apply(t(exprs(datalist.q[[i]])[,grep(levels(
        tbstat[[i]])[[type]],eval(parse(text=paste("datalist[[i]]$",colour.
        variable,sep=""))))]), 2, mean), ncol=1))
612   points(x1[1], x1[2], col=levels(classcolours[[i]])[[type]], cex=6,pch=9)}
613
614
615   export.plot(file.prefix=paste('fig',q.i,'.',an.count,'.',i,'.',figure,
        names(questions)[[q.i]],'.',analysis,names(datalist)[i],"PCA_all"),
        export.formats=export.formats.plots,height=height,width=(1 + sqrt(5))/2*
        height)
616 #}
617 figure<-figure+1
618
619 #filter####
620
621 #new filter (requires beadarray; boxplot now uses ggplot2) (libraries
      require integration with system setup)
622
623 #filter qnorm data
624 #data.fil<-list()
625 #for (i in 1:length(datalist.q)){
626   x <- pqlist[[i]]
627   mapped_probes <- mappedkeys(x)
628   # Convert to a list
629   xx <- as.list(x[mapped_probes])
630   ids <- as.character(nuID2IlluminaID(as.character(featureNames(datalist.q[[
        i]])),chipVersion=getChipInfo(datalist.q[[i]])$chipVersion[[1]]))
631   is<-intersect(names(unlist(xx)),ids)#remove probes not in database, then
        the mget call will not fail
632   qual <- unlist(mget(is, pqlist[[i]]))
633   table(qual)
634   rem <- qual == "No match" | qual == "Bad" | is.na(qual)
635   data.fil[[i]]<- datalist.q[[i]][!rem, ]
```

```
636   data.v.fil[[i]]<- datalist.v[[i]][!rem, ]
637
638 #show filtered probes ####
639
640 #filtered probes (PROBEQUALITY)
641 #for (i in 1:length(data.fil)) {
642   par(mar=c(7,5,1,1))
643   boxplot(data.fil[[i]],las=2,col=as.character(classcolours[[i]]),main=paste
      (names(datalist)[[i]],'Boxplot of selected probes'))
644   par(parbackup)
645   export.plot(file.prefix=paste('fig',q.i,'.',an.count,'.',i,'.',figure,
      names(questions)[[q.i]],'.',analysis,names(datalist)[i],'boxplot
      filtered probes'),export.formats=export.formats.plots,height=height,
      width=(1 + sqrt(5))/2*height)
646 #}
647
648 figure<-figure+1
649
650 #MDS selected probes
651
652 #for (i in 1:length(data.fil)){
653 plotSampleRelationsAD(data.fil[[i]],method="mds",color=as.character(
      classcolours[[i]]),plotchar=16)
654 legend("bottomleft",
655        levels(tbstat[[i]]),
656        pch=16,
657        col=levels(classcolours[[i]]),
658        cex=.8)
659   export.plot(file.prefix=paste('fig',q.i,'.',an.count,'.',i,'.',figure,
      names(questions)[[q.i]],'.',analysis,names(datalist)[i],"MDS_filtered"),
      export.formats=export.formats.plots,height=height,width=(1 + sqrt(5))/2*
      height)
660 #}
661 figure<-figure+1
662
663 #PCA all probes (filtered)
664
665 #for (i in 1:length(data.fil)){
666   l=colnames(exprs(data.fil[[i]]))
667   spca <- SamplePCA(exprs(data.fil[[i]]), tbstat[[i]])
668   plot(spca, col=levels(classcolours[[i]]),main=paste(nrow(exprs(data.fil[[i
      ]])),"probes (filtered)"))#,cex=2,cex.axis=2,cex.main=3,cex.lab=2.5,pch
      =16)
669   mtext(sprintf("%.2f",spca@variances[1]/sum(spca@variances)),side=1,line=3,
      adj=1,cex=1.5)
670   mtext(sprintf("%.2f",spca@variances[2]/sum(spca@variances)),side=2,line=3,
      adj=1,cex=1.5)
671 legend("bottomleft",
672        levels(tbstat[[i]]),
673        pch=16,
674        col=levels(classcolours[[i]]),
675        cex=.8)
676   # mark the group centers
677 for (type in 1:length(levels(tbstat[[i]]))){
678   x1 <- predict(spca, matrix(apply(t(exprs(data.fil[[i]])[,grep(levels(
      tbstat[[i]])[[type]],eval(parse(text=paste("datalist[[i]]$",colour.
      variable,sep="")))))]), 2, mean), ncol=1))
679   points(x1[1], x1[2], col=levels(classcolours[[i]])[[type]], cex=6,pch=9)}
680
```

```
681 export.plot(file.prefix=paste('fig',q.i,'.',an.count,'.',i,'.',figure,names(
        questions)[[q.i]],'.',analysis,names(datalist)[i],"PCA_filtered"),export
        .formats=export.formats.plots,height=height,width=(1 + sqrt(5))/2*height
        )
682 #}
683 figure<-figure+1
684 
685 #DE statistics####
686 
687 tbstat.class=list()
688 tbstat.class[[i]]<-as.factor(eval(parse(text=paste(dataset.variables[[i]],"$
        ",contrast.variable,sep=""))))
689 contrast.levels<-levels(tbstat.class[[i]])
690 
691 print(table(tbstat.class[[i]]))
692 
693 #for (i in 1:length(data.fil)){
694   design[[i]]=model.matrix(~-1+factor(tbstat.class[[i]]))
695   colnames(design[[i]])<-levels(tbstat.class[[i]])
696   design[[i]]
697   #contrast[[i]]<-makeContrasts(activeTB-notActiveTB,levels=design[[i]])
698   contrast[[i]]<-makeContrasts(
699    paste(contrast.levels[[1]],"-",contrast.levels[[2]]),
700     levels=design[[i]]
701     )
702   contrast[[i]]
703 
704 #This is not ideal: use vanill pvalue of 0.05 and deal with very large or
        small results
705 # effect.size<-pwr.t2n.test(n1=13,n2=29,d=NULL,sig.level=0.01,power=0.9,
        alternative="two.sided")$d
706 # pval<-pwr.t2n.test(n1=table(tbstat.class[[i]])[[1]],n2=table(tbstat.class
        [[i]])[[2]],d=effect.size,sig.level=NULL,power=0.9,alternative="two.
        sided")$sig.level*3
707 
708 
709 # pwr.t2n.test(n1=5,n2=13,d=effect.size,sig.level=NULL,power=0.9,alternative
        ="two.sided")$sig.level*effect.size
710 # pwr.t2n.test(n1=20,n2=31,d=effect.size,sig.level=NULL,power=0.9,
        alternative="two.sided")$sig.level*effect.size
711 
712   #Linear model: tb status
713   fit[[i]]<-lmFit(data.fil[[i]],design[[i]])
714   fit2[[i]]<-contrasts.fit(fit[[i]],contrast[[i]])
715   fit2[[i]]<-eBayes(fit2[[i]])
716   gr[[i]]<-decideTests(fit2[[i]],adjust.method="none",p.value=0.05)
717   vc[[i]]<-vennCounts(gr[[i]])
718   gr2[[i]]<-decideTests(fit2[[i]],adjust.method="fdr",p.value=0.05)
719   vc2[[i]]<-vennCounts(gr2[[i]])
720 #Venn diagrams of DE probes
721 vennDiagram(gr[[i]])
722 vennDiagram(gr2[[i]])
723 
724 #print unmodified results
725 print(paste("There are",vc[[i]][[4]],"probes significant at P=0.05 (
        unadjusted) and",vc2[[i]][[4]],"probes significant at P=0.05 (BH
        adjusted)"))
726 
727 if(sum(mapply(abs,gr2[[i]]))<300) {
```

```
728   print(paste("with p value = 0.05","there are",sum(mapply(abs,gr2[[i]])),"
        DE probes and top 300 results will be used for the remainder",sep=" "));
729   gr2[[i]]=is.element(rownames(exprs(data.fil[[i]])),as.character(
        probeID2nuID(topTable(fit2[[i]],coef=1,number=300,adjust.method="BH",
        confint=T,sort.by="B",resort.by="logFC")[[1]])[,7]))
730   }
731
732 if(sum(mapply(abs,gr2[[i]]))>2000) {
733   print(paste("with p value = 0.05","there are",sum(mapply(abs,gr2[[i]])),"
        DE probes and top 2000 results will be used for the remainder",sep=" "))
        ;
734   gr2[[i]]=is.element(rownames(exprs(data.fil[[i]])),as.character(
        probeID2nuID(topTable(fit2[[i]],coef=1,number=2000,adjust.method="BH",
        confint=T,sort.by="B",resort.by="logFC")[[1]])[,7]))}
735
736   vc2[[i]]<-vennCounts(gr2[[i]])
737
738   #Venn diagrams of DE probes
739   vennDiagram(gr[[i]])
740   vennDiagram(gr2[[i]])
741
742   #results
743   reslist[[i]]<-featureNames(data.fil[[i]][gr2[[i]]!=0,])
744   reslistTTall[[i]]<-topTable(fit2[[i]],coef=1,adjust.method="BH", confint=T
        ,sort.by="B",resort.by="logFC")
745   reslistTT30[[i]]<-topTable(fit2[[i]],coef=1,number=30,adjust.method="BH",
        confint=T,sort.by="B",resort.by="logFC")
746   reslistTT[[i]]<-topTable(fit2[[i]],coef=1,number=500,adjust.method="BH",
        confint=T,sort.by="B",resort.by="logFC")
747   reslistTT100[[i]]<-topTable(fit2[[i]],coef=1,number=100,adjust.method="BH"
        , confint=T,sort.by="B",resort.by="logFC")
748   reslistTT2000[[i]]<-topTable(fit2[[i]],coef=1,number=2000,adjust.method="
        BH", confint=T,sort.by="B",resort.by="logFC")
749   reslistTT8000[[i]]<-topTable(fit2[[i]],coef=1,number=8000,adjust.method="
        BH", confint=T,sort.by="B",resort.by="logFC")
750   reslistTT16000[[i]]<-topTable(fit2[[i]],coef=1,number=16000,adjust.method=
        "BH", confint=T,sort.by="B",resort.by="logFC")
751
752   #can also restrict by pvalue, lfc, etc.
753 #}
754
755 #volcanoplot
756 #for (i in 1:length(fit2)){
757   volcanoplot(fit2[[i]],coef=1,highlight=vc2[[i]][,2][2],cex=1,pch=20,col="
        red")
758   export.plot(file.prefix=paste('fig',q.i,'.',an.count,'.',i,'.',figure,
        names(questions)[[q.i]],'.',analysis,names(datalist)[i],'volcanoplot DE
        probes'),export.formats=export.formats.plots,height=height,width=(1 +
        sqrt(5))/2*height)
759 #}
760
761 figure<-figure+1
762
763 #show DE probes####
764
765 #heatmaps
766 #DE probes
767
768 #for (i in 1:length(data.fil)){
```

```
769   phenomatrix<-matrix(cbind(as.character(classcolours[[i]]),as.character(
        classcolours[[i]])),ncol=2)
770   superHeatmap2(x=data.fil[[i]],y=gr2[[i]]!=0,phenomatrix=phenomatrix,scale=
        "row")
771   export.plot(file.prefix=paste('fig',q.i,'.',an.count,'.',i,'.',figure,
        names(questions)[[q.i]],'.',analysis,names(datalist)[i],'heatmap DE
        probes by pval'),export.formats=export.formats.plots,height=height,width
        =(1 + sqrt(5))/2*height)
772 #}
773 figure<-figure+1
774 #top 500 by B
775
776 #for (i in 1:length(data.fil)){
777   phenomatrix<-matrix(cbind(as.character(classcolours[[i]]),as.character(
        classcolours[[i]])),ncol=2)
778
779   subset<-is.element(rownames(exprs(data.fil[[i]])),as.character(
        probeID2nuID(reslistTT[[i]][[1]])[,7]))
780
781   superHeatmap2(x=data.fil[[i]],y=subset,phenomatrix=phenomatrix,scale="row"
        )
782 export.plot(file.prefix=paste('fig',q.i,'.',an.count,'.',i,'.',figure,names(
        questions)[[q.i]],'.',analysis,names(datalist)[i],'heatmap diff probes
        top 500 by B'),export.formats=export.formats.plots,height=height,width
        =(1 + sqrt(5))/2*height);par(parbackup)
783 #}
784 figure<-figure+1
785
786 #MDS DE probes
787
788 #for (i in 1:length(data.fil)){
789   plotSampleRelationsAD(data.fil[[i]][gr2[[i]]!=0,],method="mds",color=as.
        character(classcolours[[i]]),plotchar=16)
790 legend("bottomleft",
791        levels(tbstat[[i]]),
792        pch=16,
793        col=levels(classcolours[[i]]),
794        cex=.8)
795   export.plot(file.prefix=paste('fig',q.i,'.',an.count,'.',i,'.',figure,
        names(questions)[[q.i]],'.',analysis,names(datalist)[i],"MDS_DE"),export
        .formats=export.formats.plots,height=height,width=(1 + sqrt(5))/2*height
        )
796 #}
797 figure<-figure+1
798
799 #PCA all probes (DE)
800
801 #for (i in 1:length(data.fil)){
802   l=colnames(exprs(data.fil[[i]]))
803   spca <- SamplePCA(exprs(data.fil[[i]][gr2[[i]]!=0,]), tbstat[[i]])
804   plot(spca, col=levels(classcolours[[i]]),main=paste(nrow(exprs(data.fil[[i
        ]][gr2[[i]]!=0,])),"probes"))#,cex=2,cex.axis=2,cex.main=3,cex.lab=2.5,
        pch=16)
805   mtext(sprintf("%.2f",spca@variances[1]/sum(spca@variances)),side=1,line=3,
        adj=1,cex=1.5)
806   mtext(sprintf("%.2f",spca@variances[2]/sum(spca@variances)),side=2,line=3,
        adj=1,cex=1.5)
807 legend("bottomleft",
808        levels(tbstat[[i]]),
```

```
809         pch=16,
810         col=levels(classcolours[[i]]),
811         cex=.8)
812   # mark the group centers
813 for (type in 1:length(levels(tbstat[[i]]))){
814   x1 <- predict(spca, matrix(apply(t(exprs(data.fil[[i]][gr2[[i]]!=0,])[,
        grep(levels(tbstat[[i]])[[type]],eval(parse(text=paste("datalist[[i]]$",
        colour.variable,sep=""))))]), 2, mean), ncol=1))
815   points(x1[1], x1[2], col=levels(classcolours[[i]])[[type]], cex=6,pch=9)}
816
817   export.plot(file.prefix=paste('fig',q.i,'.',an.count,'.',i,'.',figure,
        names(questions)[[q.i]],'.',analysis,names(datalist)[i],"PCA_DE"),export
        .formats=export.formats.plots,height=height,width=(1 + sqrt(5))/2*height
        )
818 #}
819 figure<-figure+1
820
821 #Export the probe lists####
822
823 dict<-nuID2IlluminaID(as.character(reslist[[i]]),species="Human")
824 dict.entrez<-nuID2EntrezID(as.character(reslist[[i]]),filterTh = NULL,lib.
      mapping='lumiHumanIDMapping', returnAllInfo = TRUE)
825 dict3<-merge(dict,dict.entrez,by.x="nuID",by.y=0)
826
827 reslistTT2000[[i]]$ProbeID<-sapply(reslistTT2000[[i]]$ProbeID,function(x){
      sprintf("%010d",as.numeric(x))})
828 full<-merge(dict3,reslistTT2000[[i]],by.x="Array_Address_Id",by.y="ProbeID",
      all.x=T)
829
830 write.csv(dict3,file=file.path(dir.results,paste(analysis,dataset.names[[i
      ]],".csv")))
831 write.csv(reslistTT[[i]],file=file.path(dir.results,paste(analysis,dataset.
      names[[i]],"_top500.csv")))
832 write.csv(full,file=file.path(dir.results,paste(analysis,dataset.names[[i]],
      "_sigWithFC.csv")))
833 write.csv(reslistTT2000[[i]],file=file.path(dir.results,paste(analysis,
      dataset.names[[i]],"_top2000.csv")))
834
835 xtabTop30<-xtable(reslistTT30[[i]],align=c(rep(c("|p{1.2cm}"),ncol(
      reslistTT30[[i]])),"|p{1cm}|"))
836
837 print(xtabTop30,hline.after=-1:nrow(reslistTT30[[i]]),include.colnames=T,
      include.rownames=F,size=c("tiny"),file=(
838   file.path(dir.results,paste("Top 30 probes for question ",q.i,": ",names(
      questions)[[q.i]],"_",dataset.names[[i]],".tex",sep=""))
839 ))
840
841 #temp closure for DE code
842 #} #opened on line 379
843
844 #Comparison
845
846 # #check overlap HIVneg and HIVpos (i.e. this is a comparison, not a
      contrast)####
847 #
848 # #this can be moved to venn script later
849 #
850 # #Venn diagram (this is actually pretty good)
851 # venndatax=list()
```

```
# venndatax[[i]]<-reslist[[i]]
# venndata500<-list()
# #for (i in 1:length(reslistTT)){
# venndata500[[i]]<-reslistTT[[i]][,2]
# #}
# venndata100<-list()
# #for (i in 1:length(reslistTT100)){
# venndata100[[i]]<-reslistTT100[[i]][,2]
# #}
# temp.venn.7<-venn.diagram(
#   x = venndatax,
#   category=c("HIVneg","HIVpos"),
#   #filename = file.path(dir.figures,"fig71_Venn_3set_lh_al_ah.tiff"),
#   #filename = file.path(dir.figures,paste("fig",figure,"_Venn_4set_gene.tiff",sep="")),
#   filename = NULL,
#   scaled = F,  ext.text = TRUE,  ext.line.lwd = 2,
#   ext.dist = -0.15,  ext.length = 0.9,  ext.pos = -4,
#   inverted = TRUE,
#   cex = 2.5,  cat.cex = 2.5,  rotation.degree = 0,
#   #main = "Overlap",  sub = "MA, VAL, TRAIN","393",
#   # main.cex = 2,  sub.cex = 1,
#   fill=c("green","red"),
#   alpha=c(.4,.4),height=3000,width=(1 + sqrt(5))/2*3000
# )
# pdf(file=file.path(dir.figures,paste("fig",figure,"_Venn_2set_probe.pdf",sep="")),height=7,width=(1 + sqrt(5))/2*7)
# grid.draw(temp.venn.7)
# dev.off()
# figure<-figure+1
#
# temp.venn.8<-venn.diagram(
#   x = venndata500,
#   category=c("HIVneg","HIVpos"),
#   #filename = file.path(dir.figures,"fig71_Venn_3set_lh_al_ah.tiff"),
#   #filename = file.path(dir.figures,paste("fig",figure,"_Venn_4set_gene.tiff",sep="")),
#   filename = NULL,
#   scaled = F,  ext.text = TRUE,  ext.line.lwd = 2,
#   ext.dist = -0.15,  ext.length = 0.9,  ext.pos = -4,
#   inverted = TRUE,
#   cex = 2.5,  cat.cex = 2.5,  rotation.degree = 0,
#   #main = "Overlap",  sub = "MA, VAL, TRAIN","393",
#   # main.cex = 2,  sub.cex = 1,
#   fill=c("green","red"),
#   alpha=c(.4,.4),height=3000,width=(1 + sqrt(5))/2*3000
# )
# pdf(file=file.path(dir.figures,paste("fig",figure,"_Venn_2set_top500_probe.pdf",sep="")),height=7,width=(1 + sqrt(5))/2*7)
# grid.draw(temp.venn.8)
# dev.off()
# figure<-figure+1
#
# temp.venn.9<-venn.diagram(
#   x = venndata100,
#   category=c("HIV neg","HIVpos"),
#   #filename = file.path(dir.figures,"fig71_Venn_3set_lh_al_ah.tiff"),
#   #filename = file.path(dir.figures,paste("fig",figure,"_Venn_4set_gene.tiff",sep="")),
```

```
#    filename = NULL,
#    scaled = F,  ext.text = TRUE,  ext.line.lwd = 2,
#    ext.dist = -0.15,  ext.length = 0.9,  ext.pos = -4,
#    inverted = TRUE,
#    cex = 2.5,  cat.cex = 2.5,  rotation.degree = 0,
#    #main = "Overlap",  sub = "MA, VAL, TRAIN","393",
#    # main.cex = 2,  sub.cex = 1,
#    fill=c("green","red"),
#    alpha=c(.4,.4),height=3000,width=(1 + sqrt(5))/2*3000
# )
# pdf(file=file.path(dir.figures,paste("fig",figure,"_Venn_2set_top100_probe
    .pdf",sep="")),height=7,width=(1 + sqrt(5))/2*7)
# grid.draw(temp.venn.9)
# dev.off()
# figure<-figure+1

#deconvolution####
analysis="Deconvolution"
an.count<-2
print("Analysis 2: Deconvolution and csDE")

#setup####
#questions:
#do I use VST data?
#do I use filtered subset for proportions and csDE?

#for (i in 1:length(datalist.v)){
  figure<-1
  #VST already done as well as filtering
  dtd<-as(datalist.v[[i]],"ExpressionSet")
  #colnamelist<-contrast.variable
  meth=list("Abbas"=gedBlood(dtd,verbose=T))
  #for (h in 1:1){
    h<-1

#deconvolution####
    decdat1<-meth[[h]]
    colname<-contrast.variable
    colnames(coef(decdat1))<-eval(parse(text=paste("dtd$",colname,sep="")))
    decdat2<-asCBC(decdat1)
    decmat<-as.matrix(coef(decdat1))
    decmat.red<-decmat[apply(decmat,1,sum)>0,]
    numbers<-as.vector(table(colnames(decmat.red)))
    names<-names(table(colnames(decmat.red)))
    nstring=paste("N=(",names[1]," ",numbers[1],",",names[2]," ",numbers[2],
    ")",sep="")


#stacked bar plot all types####
par(parbackup)
par(mar=c(8,4.5,4.5,8))
    barplot(coef(decdat1),las=2,col=brewer.pal(12,"Set3"),legend.text=
    rownames(coef(decdat1)),main=paste("Proportions of" ,nrow(coef(decdat1))
    ,"cell types",names(meth)[[h]],names(datalist)[[i]],nstring),beside=F,
    ylab="Proportion",
            args.legend=c(x=ncol(decmat.red)*1.35,y=1,cex=.7),
            cex.names=.8)
    par(parbackup)
```

```
959      export.plot(file.prefix=paste('fig',q.i,'.',an.count,'.',i,'.',figure,
         names(questions)[[q.i]],'.',analysis,names(datalist)[i],"stacked_barplot
         _all",names(meth)[[h]]),export.formats=export.formats.plots,height=
         height*1.5,width=(1 + sqrt(5))/2*height*1.5)
960      figure<-figure+1
961
962
963  #stacked barplot detected types####
964  par(parbackup)
965  par(mar=c(8,4.5,4.5,8))
966      barplot(decmat.red,las=2,col=brewer.pal(12,"Set3"),legend.text=rownames(
         decmat.red),main=paste("Proportions of" ,nrow(decmat.red),"cell types",
         names(meth)[[h]],names(datalist)[[i]],nstring),beside=F,ylab="Proportion
         ",args.legend=c(x=ncol(decmat.red)*1.35,y=1,cex=.7),cex.names=.8)
967      par(parbackup)
968      export.plot(file.prefix=paste('fig',q.i,'.',an.count,'.',i,'.',figure,
         names(questions)[[q.i]],'.',analysis,names(datalist)[i],"stacked_barplot
         _detected",names(meth)[[h]]),export.formats=export.formats.plots,height=
         height*1.5,width=(1 + sqrt(5))/2*height*1.5)
969      figure<-figure+1
970
971
972  #stacked barplot CBC####
973  par(parbackup)
974      par(mar=c(8,4.5,4.5,8))
975      barplot(coef(decdat2),las=2,col=brewer.pal(12,"Set3"),legend.text=
         rownames(coef(decdat2)),main=paste("Proportions of CBC cell types",names
         (meth)[[h]],names(datalist)[[i]],nstring),beside=F,ylab="Proportion",
         args.legend=c(x=ncol(coef(decdat2))*1.35,y=1,cex=.7),cex.names=.8)
976      par(parbackup)
977      export.plot(file.prefix=paste('fig',q.i,'.',an.count,'.',i,'.',figure,
         names(questions)[[q.i]],'.',analysis,names(datalist)[i],"stacked_barplot
          (CBC)",names(meth)[[h]]),export.formats=export.formats.plots,height=
         height*1.5,width=(1 + sqrt(5))/2*height*1.5)
978      figure<-figure+1
979
980  #stacked barplot detected types (PBMC only) ####
981      decmat.red.PBMC<-matrix(data=NA,nrow=nrow(decmat.red)-1,ncol=ncol(decmat
         .red))
982      rownames(decmat.red.PBMC)<-rownames(decmat.red[-nrow(decmat.red),])
983      colnames(decmat.red.PBMC)<-colnames(decmat.red)
984      for (l in 1:ncol(decmat.red)){
985        decmat.red.PBMC[,l]<-decmat.red[-nrow(decmat.red),l]/sum(decmat.red[-
         nrow(decmat.red),l])
986      }
987
988  #stacked barplot PBMC####
989  par(parbackup)
990      par(mar=c(8,4.5,4.5,8))
991      barplot(decmat.red.PBMC,las=2,col=brewer.pal(12,"Set3"),legend.text=
         rownames(decmat.red.PBMC),main=paste("Proportions of" ,nrow(decmat.red.
         PBMC),"PBMC cell types",names(meth)[[h]],names(datalist)[[i]],nstring),
         beside=F,ylab="Proportion",args.legend=c(x=ncol(decmat.red.PBMC)*1.35,y
         =1,cex=.7),cex.names=.8)
992      par(parbackup)
993      export.plot(file.prefix=paste('fig',q.i,'.',an.count,'.',i,'.',figure,
         names(questions)[[q.i]],'.',analysis,names(datalist)[i],"stacked_barplot
          (PBMC)",names(meth)[[h]]),export.formats=export.formats.plots,height=
         height*1.5,width=(1 + sqrt(5))/2*height*1.5)
```

```
994      figure<-figure+1
995
996  #boxplot by condition####
997
998      #convert and calculate
999      clN<-colnames(decmat.red)
1000     dN<-t(decmat.red)
1001     vN<-as.vector(dN)
1002     clvN<-rep(clN,nrow(decmat.red))
1003     typesN<-rownames(decmat.red)
1004     typevN<-list()
1005     for (j in 1:nrow(decmat.red)){typevN[[j]]=rep(typesN[[j]],ncol(decmat.
         red))}
1006     typev2N<-unlist(typevN)
1007     ndN<-data.frame(vN,clvN,typev2N)
1008     levs<-levels(as.factor(colnames(decmat.red)))
1009
1010     #plot
1011     par(mar=c(10.5,4.5,4.5,8))
1012     boxplot(vN~clvN*typev2N,data=ndN,las=2,col=c("red","green","blue","
         yellow")[1:length(levs)],main=paste("Relative abundance of cell types by
         ",colname,"in",names(datalist)[[i]],nstring,names(meth)[[h]]),varwidth=
         FALSE)
1013     legend("topleft",legend=(levs),fill=c("red","green","blue","yellow"))
1014     par(parbackup)
1015     export.plot(file.prefix=paste('fig',q.i,'.',an.count,'.',i,'.',figure,
         names(questions)[[q.i]],'.',analysis,names(datalist)[i],"boxplot_by_
         celltype",names(meth)[[h]]),export.formats=export.formats.plots,height=
         height*1.5,width=(1 + sqrt(5))/2*height*1.5)
1016     figure<-figure+1
1017
1018     #exp
1019     #par(mar=c(10.5,4.5,4.5,8))
1020     #boxplotBy(decmat.red,as.factor(colnames(decmat.red)),col=c("blue","
         yellow"),main=paste("func:boxplotBy",colnamelist[[i]],"in",names(
         datalist)[[i]],nstring,names(meth)[[h]]),scale=TRUE)
1021     #par(parbackup)
1022     #export.plot(file.prefix=paste("fig",figure,".",i,"boxplot_by_celltype",
         names(datalist)[[i]],names(meth)[[h]]),export.formats=export.formats.
         plots,height=height*1.5,width=(1 + sqrt(5))/2*height*1.5)
1023     #exp
1024     #}
1025     #}
1026
1027 #stats for comparisons####
1028     statlist<-list()
1029     for (k in 1:nrow(decmat.red)) {statlist[[k]]<-wilcox.test(
1030       decmat.red[k,colnames(decmat.red)==levs[1]],decmat.red[k,colnames(
         decmat.red)==levs[2]]
1031     )$p.value}
1032     tests<-data.frame("names"=rownames(decmat.red),"pval"=as.vector(unlist(
         statlist)),"man_B"=as.vector(unlist(statlist))*nrow(decmat.red),"bonf"=p
         .adjust(as.vector(unlist(statlist)),method="bonferroni"),"BH"=p.adjust(
         as.vector(unlist(statlist)),method="BH"))
1033     write.csv(tests,file=file.path(dir.results,paste("Stats_",q.i,'.',an.
         count,'.',i,"_",names(datalist)[[i]],names(meth)[[h]],"_",colname,"_
         wilcox.csv",sep="")))
1034
1035
```

```
1036
1037 #boxplot by condition for matched blood/fluid for proportions normalised as
       PBMCs (i.e. neutrophils removed)####
1038     #convert
1039
1040     clN<-colnames(decmat.red.PBMC)
1041     dN<-t(decmat.red.PBMC)
1042     vN<-as.vector(dN)
1043     clvN<-rep(clN,nrow(decmat.red.PBMC))
1044     typesN<-rownames(decmat.red.PBMC)
1045     typevN<-list()
1046     for (j in 1:nrow(decmat.red.PBMC)){typevN[[j]]=rep(typesN[[j]],ncol(
       decmat.red.PBMC))}
1047     typev2N<-unlist(typevN)
1048     ndN<-data.frame(vN,clvN,typev2N)
1049     levs<-levels(as.factor(colnames(decmat.red.PBMC)))
1050     par(mar=c(10.5,4.5,4.5,8))
1051     boxplot(vN~clvN*typev2N,data=ndN,las=2,col=c("red","green","blue","
       yellow")[1:length(levs)],main=paste("Relative abundance of PBMC types by
       ",colname,"in",names(datalist)[[i]],nstring,names(meth)[[h]]))
1052     legend("topleft",legend=(levs),fill=c("red","green","blue","yellow"))
1053     par(parbackup)
1054     export.plot(file.prefix=paste('fig',q.i,'.',an.count,'.',i,'.',figure,
       names(questions)[[q.i]],'.',analysis,names(datalist)[i],"boxplot_by_PBMC
       _celltype",names(meth)[[h]]),export.formats=export.formats.plots,height=
       height*1.5,width=(1 + sqrt(5))/2*height*1.5)
1055     figure<-figure+1
1056
1057 #stats for comparisons####
1058     statlist<-list()
1059     for (k in 1:nrow(decmat.red.PBMC)) {statlist[[k]]<-wilcox.test(
1060       decmat.red.PBMC[k,colnames(decmat.red.PBMC)==levs[1]],decmat.red.PBMC[
       k,colnames(decmat.red.PBMC)==levs[2]]
1061     )$p.value}
1062     #tests.PBMC<-data.frame("names"=rownames(decmat.red.PBMC),"pval"=as.
       vector(unlist(statlist))*nrow(decmat.red.PBMC))#bonferroni
1063     tests.PBMC<-data.frame("names"=rownames(decmat.red.PBMC),"pval"=as.
       vector(unlist(statlist)),"man_B"=as.vector(unlist(statlist))*nrow(decmat
       .red),"bonf"=p.adjust(as.vector(unlist(statlist)),method="bonferroni"),"
       BH"=p.adjust(as.vector(unlist(statlist)),method="BH"))
1064     write.csv(tests.PBMC,file=file.path(dir.results,paste("Stats_PBMC_",q.i,
       '.',an.count,'.',i,"_",names(datalist)[[i]],names(meth)[[h]],"_",colname
       ,"_wilcox.csv",sep="")))
1065
1066 #csde ####
1067
1068     #PC added to B cells
1069     th<-apply(coef(decdat1)[1:2,],2,sum)
1070     tc<-apply(coef(decdat1)[3:4,],2,sum)
1071     b<-apply(coef(decdat1)[5:10,],2,sum)
1072     #pc<-coef(decdat1)[10,]
1073     nk<-apply(coef(decdat1)[11:12,],2,sum)
1074     mo<-apply(coef(decdat1)[13:14,],2,sum)
1075     dc<-apply(coef(decdat1)[15:16,],2,sum)
1076     neut<-coef(decdat1)[17,]
1077
1078     props<-matrix(
1079       rbind(th,tc,b,nk,mo,dc,neut
1080       ),nrow=7)
```

```
1081      rownames(props)<-c("th","tc","b","nk","mo","dc","neut")
1082      colnames(props)<-colnames(coef(decdat1))
1083
1084      #Smallprops
1085      l<-apply(coef(decdat1)[1:10,],2,sum)
1086      nk<-apply(coef(decdat1)[11:12,],2,sum)
1087      apc<-apply(coef(decdat1)[13:16,],2,sum)
1088      neut<-coef(decdat1)[17,]
1089
1090      smallprops<-matrix(
1091        rbind(l,nk,apc,neut
1092        ),nrow=4)
1093      rownames(smallprops)<-c("lym","nk","apc","neut")
1094      colnames(smallprops)<-colnames(coef(decdat1))
1095
1096      subset<-is.element(rownames(exprs(datalist.v[[i]])),as.character(
          probeID2nuID(reslistTT[[i]][[1]])[,7]))
1097      subset2<-is.element(rownames(exprs(datalist.v[[i]])),as.character(
          targetID2nuID(featureData(Abbas)$SYMBOL,lib='lumiHumanIDMapping')))
1098      subset3<-is.element(nuID2targetID(rownames(exprs(datalist.v[[i]]))),
          featureData(Abbas)$SYMBOL)
1099      subset4<-is.element(rownames(exprs(datalist.v[[i]])),as.character(
          probeID2nuID(reslistTT2000[[i]][[1]])[,7]))
1100      subset5<-is.element(rownames(exprs(datalist.v[[i]])),as.character(
          probeID2nuID(reslistTT8000[[i]][[1]])[,7]))
1101
1102  if(min(table(tbstat.class[[i]]))<nrow(props)) {
1103    props<-smallprops
1104    print("smallprops used")} else {print("props used")}
1105
1106      #cs.de<-ged(dtd[subset|subset2,],coef(decdat1),data=as.factor(colnames(
          coef(decdat1))),verbose=TRUE,nperm=1000)
1107      #cs.de<-ged(dtd,props,data=as.factor(colnames(coef(decdat1))),verbose=
          TRUE,nperm=1000)
1108     # rm(cs.de)
1109  cs.de<-c(1,2,3)
1110  plot(x=1:3,y=1:3,main="csDE has failed!")
1111      try(
1112      cs.de<-ged(
1113        dtd[subset|subset3,],
1114        props,
1115        data=as.factor(colnames(props)),
1116        verbose=TRUE,nperm=1000))
1117      par(parbackup);
1118      try(csplot(cs.de));
1119      export.plot(file.prefix=paste('fig',q.i,'.',an.count,'.',i,'.',figure,
          names(questions)[[q.i]],'.',analysis,names(datalist)[i],"csDE_FDR_by_
          celltype",names(datalist)[[i]],names(meth)[[h]]),export.formats=export.
          formats.plots,height=height*1.5,width=(1 + sqrt(5))/2*height*1.5);
1120      figure<-figure+1
1121
1122      tt<-"no csDE result"
1123      try(tt<-csTopTable(cs.de,decreasing=F))
1124  if(tt!="no csDE result"){
1125      names(tt$neut)
1126      dict.th<-nuID2IlluminaID(as.character(names(tt$th)),species="Human")
1127      write.csv(dict.th,file=file.path(dir.results,paste("csDE_Th_",q.i,'.',an
          .count,'.',i,"_",names(datalist)[[i]],names(meth)[[h]],"_",colname,".csv
          ",sep="")))
```

```
1128      dict.tc<-nuID2IlluminaID(as.character(names(tt$tc)),species="Human")
1129      write.csv(dict.tc,file=file.path(dir.results,paste("csDE_Tc_",q.i,'.',an
          .count,'.',i,"_",names(datalist)[[i]],names(meth)[[h]],"_",colname,".csv
          ",sep="")))
1130      dict.b<-nuID2IlluminaID(as.character(names(tt$b)),species="Human")
1131      write.csv(dict.b,file=file.path(dir.results,paste("csDE_B_",q.i,'.',an.
          count,'.',i,"_",names(datalist)[[i]],names(meth)[[h]],"_",colname,".csv"
          ,sep="")))
1132      dict.nk<-nuID2IlluminaID(as.character(names(tt$nk)),species="Human")
1133      write.csv(dict.nk,file=file.path(dir.results,paste("csDE_NK_",q.i,'.',an
          .count,'.',i,"_",names(datalist)[[i]],names(meth)[[h]],"_",colname,".csv
          ",sep="")))
1134      dict.mo<-nuID2IlluminaID(as.character(names(tt$mo)),species="Human")
1135      write.csv(dict.mo,file=file.path(dir.results,paste("csDE_MO_",q.i,'.',an
          .count,'.',i,"_",names(datalist)[[i]],names(meth)[[h]],"_",colname,".csv
          ",sep="")))
1136      dict.dc<-nuID2IlluminaID(as.character(names(tt$dc)),species="Human")
1137      write.csv(dict.dc,file=file.path(dir.results,paste("csDE_DC_",q.i,'.',an
          .count,'.',i,"_",names(datalist)[[i]],names(meth)[[h]],"_",colname,".csv
          ",sep="")))
1138      dict.neut<-nuID2IlluminaID(as.character(names(tt$neut)),species="Human")
1139      write.csv(dict.neut,file=file.path(dir.results,paste("csDE_neut_",q.i,'.
          ',an.count,'.',i,"_",names(datalist)[[i]],names(meth)[[h]],"_",colname,"
          .csv",sep="")))
1140      dict.lym<-nuID2IlluminaID(as.character(names(tt$lym)),species="Human")
1141      write.csv(dict.lym,file=file.path(dir.results,paste("csDE_lym_",q.i,'.',
          an.count,'.',i,"_",names(datalist)[[i]],names(meth)[[h]],"_",colname,".
          csv",sep="")))
1142      dict.apc<-nuID2IlluminaID(as.character(names(tt$apc)),species="Human")
1143      write.csv(dict.apc,file=file.path(dir.results,paste("csDE_apc_",q.i,'.',
          an.count,'.',i,"_",names(datalist)[[i]],names(meth)[[h]],"_",colname,".
          csv",sep="")))} else print(tt)
1144
1145  #  }methods in deconvolution for loop
1146  #}deconvoloution for loop
1147
1148  #wgcna####
1149  analysis="wgcna"
1150  an.count<-3
1151  print("Analysis 3: WGCNA")
1152
1153  #setup####
1154  #for (i in 1:length(datalist)){
1155    #reset figure
1156    figure<-1
1157  #select data for WGCNA: use top 8000 DE genes
1158  subset<-is.element(rownames(exprs(data.fil[[i]])),as.character(probeID2nuID(
          reslistTT8000[[i]][[1]])[,7]))
1159
1160  usedata<-data.fil[[i]][subset,]
1161  datExpr0 = data.frame(t(exprs(usedata)))
1162
1163  # Take a quick look at what is in the data set:
1164  dim(datExpr0);
1165  names(datExpr0)[1:10];#probes
1166  rownames(datExpr0)#samples
1167
1168  #1.b Checking data for excessive missing values and identification of
          outlier microarray samples####
```

```
#check for missing values
gsg=goodSamplesGenes(datExpr0,verbose=3);
gsg$allOK
#remove offending samples
if (!gsg$allOK)
{
  # Optionally, print the gene and sample names that were removed:
  if (sum(!gsg$goodGenes)>0)
    printFlush(paste("Removing genes:", paste(names(datExpr0)[!gsg$goodGenes
    ], collapse = ", ")));
  if (sum(!gsg$goodSamples)>0)
    printFlush(paste("Removing samples:", paste(rownames(datExpr0)[!gsg$
    goodSamples], collapse = ", ")));
  # Remove the offending genes and samples from the data:
  datExpr0 = datExpr0[gsg$goodSamples, gsg$goodGenes]
}

#cluster samples to look for outliers####
sampleTree = flashClust(dist(datExpr0), method = "average");
# Plot the sample tree: Open a graphic output window of size 12 by 9 inches
# The user should change the dimensions if the window is too large or too
    small.
#sizeGrWindow(12,9)
#pdf(file = "Plots/sampleClustering.pdf", width = 12, height = 9);
par(cex = 0.6);
par(mar = c(0,4,2,0))
plot(sampleTree, main = "Sample clustering to detect outliers", sub="", xlab
    ="", cex.lab = 1.5,
     cex.axis = 1.5, cex.main = 2)
# Plot a line to show the cut
abline(h = 80, col = "red")
export.plot(file.prefix=paste('fig',q.i,'.',an.count,'.',i,'.',figure,names(
    questions)[[q.i]],'.',analysis,names(datalist)[i],'outlier plot all
    samples'),export.formats=export.formats.plots,height=height,width=(1 +
    sqrt(5))/2*height)
par(parbackup)
figure<-figure+1

# Determine cluster under the line
clust = cutreeStatic(sampleTree, cutHeight = 80, minSize = 10)
table(clust)
# clust 1 contains the samples we want to keep.
keepSamples = (clust==1)
datExpr = datExpr0[keepSamples, ]
nGenes = ncol(datExpr)
nSamples = nrow(datExpr)

#removing outliers breaks the code further down. For now, don't remove
    outliers at all
datExpr=datExpr0

#1.c Prepare clinical trait data#####
traitData = pData(usedata);
dim(traitData)
names(traitData)

# remove columns that hold information we do not need.
base.traits<-c(2,7,19,45,47,9,10,11,12,13,14,15,16,17,18,20,22,38)

```

```
allTraits = traitData[, c(base.traits,extra.traits)]
#allTraits = traitData
#allTraits = allTraits[, c(2, 11:36) ];
dim(allTraits)
names(allTraits)

pcSamples = rownames(datExpr);
traitRows = match(pcSamples, allTraits$sample_name);
datTraits = allTraits[traitRows, -1];
rownames(datTraits)<-traitData$sample_name

datTraits<-data.frame(apply(datTraits,2, function(x){replace(x, x =="",NA)})
    )

datTraits[, c(1,2,3,4,5,13,14,17)] <- lapply(datTraits[, c
    (1,2,3,4,5,13,14,17)], as.factor)
datTraits[, c(1,2,3,4,5,13,14,17)] <- lapply(datTraits[, c
    (1,2,3,4,5,13,14,17)], as.numeric)

datTraits<-as.matrix(datTraits)
datTraits<-apply(datTraits,2,as.numeric)

#datTraits<-data.frame(lapply(datTraits,as.numeric))
rownames(datTraits) = allTraits[traitRows, 1];
collectGarbage();
datTraits <- datTraits[,colSums(is.na(datTraits))<nrow(datTraits)]#this
    removes all columns where all values are NA
uniquelength <- apply(datTraits,2,function(x) length(unique(x[!is.na(x)])))

datTraits <- subset(datTraits, select=uniquelength>1)#this removes all
    columns where all values are the same after removing NAs
datTraits<-data.frame(datTraits)
#expression data and phenotype data are now in analagous data frames

#Visualisation of how the traits relate to the sample dendrogram####
# Re-cluster samples
sampleTree2 = flashClust(dist(datExpr), method = "average")

# Convert traits to a color representation: white means low, red means high,
     grey means missing entry
traitColors = numbers2colors(as.matrix(datTraits), signed = FALSE);

# Plot the sample dendrogram and the colors underneath.
plotDendroAndColors(sampleTree2, traitColors,cex.dendroLabels = 0.5,cex.
    colorLabels=.5,cex.lab=.7,cex.axis=.7,
                    groupLabels = colnames(datTraits),
                    main = "Sample dendrogram and trait heatmap")
export.plot(file.prefix=paste('fig',q.i,'.',an.count,'.',i,'.',figure,names(
    questions)[[q.i]],'.',analysis,names(datalist)[i],'samples_trait_combo_
    plot'),export.formats=export.formats.plots,height=height,width=(1 + sqrt
    (5))/2*height)
par(parbackup)
figure<-figure+1

#2.c.1 Choosing the soft-thresholding power: analysis of network topology
    ####
# Choose a set of soft-thresholding powers
powers = c(c(1:10), seq(from = 12, to=20, by=2))
# Call the network topology analysis function
```

```
sft = pickSoftThreshold(datExpr, powerVector = powers, RsquaredCut = 0.80, verbose = 5)
# Plot the results:
#sizeGrWindow(9, 5)
par(mfrow = c(1,2));par(mar=c(4,4,2,2))
cex1 = 0.9;
# Scale-free topology fit index as a function of the soft-thresholding power
plot(sft$fitIndices[,1], -sign(sft$fitIndices[,3])*sft$fitIndices[,2],
     xlab="Soft Threshold (power)",ylab="Scale Free Topology Model Fit, signed R^2",type="n",
     main = paste("Scale independence"));
text(sft$fitIndices[,1], -sign(sft$fitIndices[,3])*sft$fitIndices[,2],
     labels=powers,cex=cex1,col="red");
# this line corresponds to using an R^2 cut-off of h
abline(h=0.80,col="red")
# Mean connectivity as a function of the soft-thresholding power
plot(sft$fitIndices[,1], sft$fitIndices[,5],
     xlab="Soft Threshold (power)",ylab="Mean Connectivity", type="n",
     main = paste("Mean connectivity"))
text(sft$fitIndices[,1], sft$fitIndices[,5], labels=powers, cex=cex1,col="red")
par(mfrow = c(1,1));

export.plot(file.prefix=paste('fig',q.i,'.',an.count,'.',i,'.',figure,names(questions)[[q.i]],'.',analysis,names(datalist)[i],'scale_independence_and _mean_connectivity'),export.formats=export.formats.plots,height=height,width=(1 + sqrt(5))/2*height)
par(parbackup)
figure<-figure+1
print(paste("Soft-thresholding power:",sft$powerEstimate ))

#2.c.2 Block-wise network construction and module detection####
thispower = sft$powerEstimate
if (is.na(thispower)){
  thispower=min(sft$fitIndices[sft$fitIndices$SFT.R.sq>0.75,]$Power)
  print("No power estimate for R squared of .80, using .75")
  print(paste("New soft-thresholding power:",thispower ))}

bwnet = blockwiseModules(datExpr, maxBlockSize = 8000,
                         power = thispower,#networkType='signed',TOMType="signed",
                         minModuleSize = 20,
                         reassignThreshold = 0, mergeCutHeight = 0.25,
                         numericLabels = TRUE,
                         saveTOMs = FALSE,
                         saveTOMFileBase = "TBPC_TOM_blockwise_10K",
                         verbose = 3)
#xxx

# Convert labels to colors for plotting
mergedColors = labels2colors(bwnet$colors)
# Plot the dendrogram and the module colors underneath
plotDendroAndColors(bwnet$dendrograms[[1]], mergedColors[bwnet$blockGenes[[1]]],
                    "Module colors",
                    dendroLabels = FALSE, hang = 0.03,
                    addGuide = TRUE, guideHang = 0.05,
                    main=paste(names(datalist[[1]]),nrow(datExpr),"samples",ncol(datExpr),"probes")
```

```
                    )
export.plot(file.prefix=paste('fig',q.i,'.',an.count,'.',i,'.',figure,names(
    questions)[[q.i]],'.',analysis,names(datalist)[i],'dendrogram_and_module
    _colours'),export.formats=export.formats.plots,height=height,width=(1 +
    sqrt(5))/2*height)
par(parbackup)
figure<-figure+1

#plotColorUnderTree(bwnet$dendrograms[[1]], mergedColors[bwnet$blockGenes
    [[1]]])
#save the tree (module assignment and module eigengene information)
bwmoduleLabels = bwnet$colors
bwmoduleColors = labels2colors(bwnet$colors)
bwMEs = bwnet$MEs;
bwgeneTree = bwnet$dendrograms[[1]];
# Relabel blockwise modules
bwLabels = matchLabels(bwnet$colors, bwmoduleLabels);
# Convert labels to colors for plotting
bwModuleColors = labels2colors(bwLabels)

print("number of genes in each module")
print(table(bwLabels))

#3. Relating modules to external information and identifying important genes
    ####

#3.a Quantifying module trait associations####

# Define numbers of genes and samples
nGenes = ncol(datExpr);
nSamples = nrow(datExpr);
# Recalculate MEs with color labels
MEs0 = moduleEigengenes(datExpr, bwmoduleColors)$eigengenes
MEs = orderMEs(MEs0)
moduleTraitCor = cor(MEs, datTraits, use = "p");
moduleTraitPvalue = corPvalueStudent(moduleTraitCor, nSamples);

# Will display correlations and their p-values
textMatrix = paste(signif(moduleTraitCor, 2), "\n(",
                   signif(moduleTraitPvalue, 1), ")", sep = "");
dim(textMatrix) = dim(moduleTraitCor)

par(mar = c(5,9,2,2));par(mfrow=c(1,1))
# Display the correlation values within a heatmap plot
labeledHeatmap(Matrix = moduleTraitCor,
               xLabels = names(datTraits),
               yLabels = names(MEs),
               ySymbols = names(MEs),
               colorLabels = FALSE,
               colors = greenWhiteRed(50),
               textMatrix = textMatrix,
               setStdMargins = FALSE,
               cex.text = 0.5,
               zlim = c(-1,1),
               main = paste("Module-trait relationships"))
export.plot(file.prefix=paste('fig',q.i,'.',an.count,'.',i,'.',figure,names(
    questions)[[q.i]],'.',analysis,names(datalist)[i],'module-trait_
    relationships'),export.formats=export.formats.plots,height=height,width
    =(1 + sqrt(5))/2*height)
```

```
1369 par(parbackup)
1370 figure<-figure+1
1371
1372 # clustered version, with NA columns removed
1373
1374 par(mar = c(2,2,2,5));par(cex.main=.5);
1375 heatmap(moduleTraitCor[,!apply(moduleTraitCor,2,is.na)[1,]],col=
         greenWhiteRed(50),main="Clustered module-trait relationships",mar=c(6,1)
         ,cexRow=.5,cexCol=.5)
1376 export.plot(file.prefix=paste('fig',q.i,'.',an.count,'.',i,'.',figure,names(
         questions)[[q.i]],'.',analysis,names(datalist)[i],'clust_module-trait_
         relationships'),export.formats=export.formats.plots,height=height,width
         =(1 + sqrt(5))/2*height)
1377 par(parbackup)
1378 figure<-figure+1
1379
1380 #3.b Gene relationship to trait and important modules: Gene Significance and
          Module Membership####
1381
1382 #We quantify associations of individual genes with our trait of interest (
         which depends on the question/ contrast) by defining Gene Significance
         GS
1383 #as (the absolute value of) the correlation between the gene and the trait.
         For each module, we also define a
1384 #quantitative measure of module membership MM as the correlation of the
         module eigengene and the gene expression
1385 #profile. This allows us to quantify the similarity of all genes on the
         array to every module.
1386
1387 #in this context 'class' refers to the contrast variable!
1388 class=as.data.frame(eval(parse(text=paste("datTraits$",contrast.variable))))
1389 names(class)=contrast.variable
1390
1391 # names (colors) of the modules
1392 modNames = substring(names(MEs), 3)
1393 geneModuleMembership = as.data.frame(cor(datExpr, MEs, use = "p"));
1394 MMPvalue = as.data.frame(corPvalueStudent(as.matrix(geneModuleMembership),
         nSamples));
1395 names(geneModuleMembership) = paste("MM", modNames, sep="");
1396 names(MMPvalue) = paste("p.MM", modNames, sep="");
1397
1398 geneTraitSignificance = as.data.frame(cor(datExpr, class, use = "p"));
1399 GSPvalue = as.data.frame(corPvalueStudent(as.matrix(geneTraitSignificance),
         nSamples));
1400 names(geneTraitSignificance) = paste("GS.", names(class), sep="");
1401 names(GSPvalue) = paste("p.GS.", names(class), sep="");
1402
1403 #3.c Intramodular analysis: identifying genes with high GS and MM####
1404
1405 #Using the GS and MM measures, we can identify genes that have a high
         significance for class as well as high module membership in interesting
         modules. We plot a scatterplot of Gene Significance vs. Module
         Membership in all modules:
1406
1407 par(parbackup);
1408 for (mods in modNames){
1409 module = mods
1410 column = match(module, modNames);
1411 moduleGenes = bwmoduleColors==module;
```

```
1412
1413 if(length(abs(geneModuleMembership[moduleGenes, column]))>19){
1414
1415 verboseScatterplot(abs(geneModuleMembership[moduleGenes, column]),
1416                    abs(geneTraitSignificance[moduleGenes, 1]),
1417                    abline=T,abline.color = "red",
1418                    xlab = paste("Module Membership in", module, "module"),
1419                    ylab = paste("Gene significance for",contrast.variable),
1420                    main = paste(mods," module membership vs. gene
       significance\n"),
1421                    cex.main = 1.2, cex.lab = 1.2, cex.axis = 1.2, col =
       module)
1422 export.plot(file.prefix=paste('fig',q.i,'.',an.count,'.',i,'.',figure,names(
       questions)[[q.i]],'.',analysis,names(datalist)[i],'MM_vs_GS_',module),
       export.formats=export.formats.plots,height=height,width=(1 + sqrt(5))/2*
       height)
1423 par(parbackup)
1424 }
1425
1426 figure<-figure+1}#moved figure out of for loop!
1427
1428 #3.d Summary output of network analysis results####
1429
1430 #retrieve various annotations to attach to feature and module data
1431 features<-featureNames(usedata)
1432 annotSYM<-as.character(getSYMBOL(features,"lumiHumanAll.db"))
1433 annotACC<-as.character(lookUp(features,"lumiHumanAll.db","ACCNUM"))
1434 annotCHR<-as.character(lookUp(features,"lumiHumanAll.db","CHR"))
1435 annotCHRLOC<-as.character(lookUp(features,"lumiHumanAll.db","CHRLOC"))
1436 annotENSEMBL<-as.character(lookUp(features,"lumiHumanAll.db","ENSEMBL"))
1437 annotENTREZID<-as.character(lookUp(features,"lumiHumanAll.db","ENTREZID"))
1438 annotGO<-as.character(lookUp(features,"lumiHumanAll.db","GO"))
1439 annotREFSEQ<-as.character(lookUp(features,"lumiHumanAll.db","REFSEQ"))
1440 annotUNIPROT<-as.character(lookUp(features,"lumiHumanAll.db","UNIPROT"))
1441
1442 annot<-data.frame(cbind(features,annotSYM,annotACC,annotCHR,annotCHRLOC,
       annotENSEMBL,annotENTREZID,annotGO,annotREFSEQ,annotUNIPROT))
1443 ##-- need to do this manually:
1444 dim(annot)
1445 names(annot)
1446 probes = names(datExpr)
1447 probes2annot = match(probes, annot$features)
1448 # The following is the number or probes without annotation:
1449 sum(is.na(probes2annot))
1450 # Should return 0.It doesn't. Not a surprise.
1451
1452 # Create the starting data frame
1453 geneInfo0 = data.frame(substanceBXH = probes,
1454                        geneSymbol = annot$annotSYM[probes2annot],
1455                        LocusLinkID = annot$annotENTREZID[probes2annot],
1456                        moduleColor = bwmoduleColors,
1457                        geneTraitSignificance,
1458                        GSPvalue)
1459 # Order modules by their significance for class
1460 modOrder = order(-abs(cor(MEs, class, use = "p")));
1461 # Add module membership information in the chosen order
1462 for (mod in 1:ncol(geneModuleMembership)){
1463   oldNames = names(geneInfo0)
1464   geneInfo0 = data.frame(geneInfo0, geneModuleMembership[, modOrder[mod]],
```

```
                          MMPvalue[, modOrder[mod]]);
  names(geneInfo0) = c(oldNames, paste("MM.", modNames[modOrder[mod]], sep="
    "),
                       paste("p.MM.", modNames[modOrder[mod]], sep=""))
}
# Order the genes in the geneInfo variable first by module color, then by
    geneTraitSignificance
geneOrder = order(geneInfo0$moduleColor, -abs(eval(parse(text=paste("
    geneInfo0$GS.",contrast.variable,sep=""))))));
geneInfo = geneInfo0[geneOrder, ]
write.csv(geneInfo,file=file.path(dir.results,paste('table',q.i,'.',an.count
    ,'.',i,'.',names(questions)[[q.i]],'.',analysis,names(datalist)[i],"
    geneInfoClass.csv")))

#4 Interfacing network analysis with other data such as functional
    annotation and gene ontology####

#4.a Output gene lists for use with online software and services####

# Match probes in the data set to the probe IDs in the annotation file
probes = names(datExpr)
probes2annot = match(probes, annot$features)
# Get the corresponding Locus Link IDs
allLLIDs = as.numeric(annot$annotENTREZID[probes2annot]);
allLLIDs=allLLIDs[!is.na(allLLIDs)]
# $ Choose interesting modules
intModules = unique(bwmoduleColors)

for (module in intModules){
  # Select module probes
  modGenes = (bwmoduleColors==module)
  # Get their entrez ID codes
  modLLIDs = allLLIDs[modGenes];
  # Write them into a file
  fileName = file.path(dir.results,paste('LLID',q.i,'.',an.count,'.',i,'.',
    names(datalist)[[i]],"_LocusLinkIDs-", module, ".txt",sep=""));
  write.table(as.data.frame(modLLIDs), file = fileName,row.names = FALSE,
    col.names = FALSE)
}

# As background in the enrichment analysis, we will use all probes in the
    analysis.
fileName = file.path(dir.results,paste('ALL-LLID',q.i,'.',an.count,'.',i,'.'
    ,names(datalist)[[i]],"_entrez_ids-all.txt"));
write.table(as.data.frame(allLLIDs), file = fileName,
            row.names = FALSE, col.names = FALSE)

#4.b Enrichment analysis directly within R####
GOenr = GOenrichmentAnalysis(bwmoduleColors, allLLIDs, organism = "human",
    nBestP = 20,includeOffspring=FALSE);
tab = GOenr$bestPTerms[[4]]$enrichment
names(tab)
write.table(tab,file=file.path(dir.results,paste('GO',q.i,'.',an.count,'.',i
    ,'.',names(datalist)[[i]],"_GOEnrichmentTable.csv")),sep=",",quote=TRUE,
    row.names=FALSE)

#on-screen
keepCols = c(1, 2, 5, 6, 7, 12, 13);
screenTab = tab[, keepCols];
```

```
# Round the numeric columns to 2 decimal places:
numCols = c(3, 4);
screenTab[, numCols] = signif(apply(screenTab[, numCols], 2, as.numeric), 2)
# Truncate the the term name to at most 40 characters
screenTab[, 7] = substring(screenTab[, 7], 1, 40)
# Shorten the column names:
colnames(screenTab) = c("module", "size", "p-val", "Bonf", "nInTerm", "ont",
     "term name");
rownames(screenTab) = NULL;
# Set the width of R output. The reader should play with this number to
    obtain satisfactory output.
options(width=95)
# Finally, display the enrichment table:
screenTab

#5. Network visualization using WGCNA functions####

#5a. make TOMplot#only run this if lots of memory. don't include in RData
    ####

#problems:
#1. not complete network, but only one block at a time
#2. despite this, it is too big

# Calculate topological overlap anew: this could be done more efficiently by
     saving the TOM

#get hold of TOMs inRData files in ~

# calculated during module detection, but let us do it again here. #!No!
#dissTOM = 1-TOMsimilarityFromExpr(datExpr, power = 11);

# do this for eachj block, i.e.TOM?

# ##don't do TOM plots at this stage
# load("TBPC_TOM_blockwise_10K-block.1.RData")
# TOM<-as.matrix(TOM)
# dissTOM1=1-TOM
# # Transform dissTOM with a power to make moderately strong connections
    more visible in the heatmap
# plotTOM1 = dissTOM1^7;
# # Set diagonal to NA for a nicer plot
# #diag(plotTOM1) = NA;
# # Call the plot function
# #sizeGrWindow(9,9)
#
# ##-- can't plot whole tom due to memory constraints...
# plotDendroAndColors(bwnet$dendrograms[[1]], bwModuleColors[bwnet$
    blockGenes[[1]]],
#                     "Module colors", main = "Gene dendrogram and module
    colors in block 1",
#                     dendroLabels = FALSE, hang = 0.03,
#                     addGuide = TRUE, guideHang = 0.05)
# TOMplot(plotTOM1,bwnet$dendrograms[[1]], bwModuleColors[bwnet$blockGenes
    [[1]]], main = "Network heatmap plot, all genes, block1")
#
# #Plot only some genes..
#
# nSelect = 1500
```

```
# # For reproducibility, we set the random seed
# set.seed(10);
# select = sample(dim(plotTOM1)[[1]], size = nSelect);
# selectTOM = dissTOM1[select, select];
# # There is no simple way of restricting a clustering tree to a subset of genes, so we must re-cluster.
# selectTree = flashClust(as.dist(selectTOM), method = "average")
# selectColors = bwmoduleColors[select];
# # Open a graphical window
# #sizeGrWindow(9,9)
# # Taking the dissimilarity to a power, say 10, makes the plot more informative by effectively changing
# # the color palette; setting the diagonal to NA also improves the clarity of the plot
# plotDiss = selectTOM^7;
# diag(plotDiss) = NA;
# TOMplot(plotDiss, selectTree, selectColors, main = "Network heatmap plot, selected genes")
# rm(TOM)

#5.b Visualizing the network of eigengenes####
# Recalculate module eigengenes
MEs = moduleEigengenes(datExpr, bwmoduleColors)$eigengenes
# Isolate numeric traits from the clinical traits and add the traits to existing module eigengenes

MEtraits<-list()
for (trait in 1:length(datTraits)){
  tmp<-as.data.frame(datTraits[[trait]])
  names(tmp)<-names(datTraits[trait])
  string=paste("ME network for ",names(datTraits[trait]),sep="")
  data=orderMEs(cbind(MEs,tmp))
  MEtraits[[trait]]=data
  names(MEtraits)[[trait]]=string
}

# Plot the relationships among the eigengenes and the trait
par(cex = 0.9)
for (m in 1:length(MEtraits)){
plotEigengeneNetworks(MEtraits[[m]],names(MEtraits)[[m]], marDendro = c(0,3,1,5), marHeatmap = c(3,4,1,1), cex.lab = 0.8, xLabelsAngle = 90)
export.plot(file.prefix=paste('fig',q.i,'.',an.count,'.',i,'.',figure,names(questions)[[q.i]],'.',analysis,names(datalist)[i],m,names(MEtraits)[[m]]),export.formats=export.formats.plots,height=height,width=(1 + sqrt(5))/2*height)
par(parbackup)
}
figure<-figure+1

#6 Exporting network data to network visualization software####

#6.a Exporting to Cytoscape

# Recalculate topological overlap if needed
TOM = TOMsimilarityFromExpr(datExpr, power = thispower);
# Read in the annotation file - #_# not needed
#annot = read.csv(file =file.path(dir.results,"GeneAnnotation.csv"));
# Select modules

```

```
modules = intModules
probesM = features[bwnet$blockGenes[[1]]]
# Select module probes
for (cytmod in modules){
inModule = is.finite(match(bwModuleColors[bwnet$blockGenes[[1]]], cytmod));
modProbes = probesM[inModule];
modGenes = annot$annotSYM[match(modProbes, annot$features)];
# Select the corresponding Topological Overlap
modTOM = TOM[inModule, inModule];
dimnames(modTOM) = list(modProbes, modProbes)
# Export the network into edge and node list files Cytoscape can read
setwd(dir.results)

exportNetworkToCytoscape(modTOM,
                                edgeFile = paste('CS',q.i,'.',an.count,'.',i,'.',names(datalist)[[i]],"CS-edges-", paste(cytmod, collapse="-"), ".txt", sep=""),
                                nodeFile = paste('CS',q.i,'.',an.count,'.',i,'.',names(datalist)[[i]],"CS-nodes-", paste(cytmod, collapse="-"), ".txt", sep=""),
                                weighted = TRUE,
                                threshold = 0.02,
                                nodeNames = modProbes,
                                altNodeNames = modGenes,
                                nodeAttr = bwmoduleColors[bwnet$blockGenes[[1]]][inModule]);
rm(modTOM)
collectGarbage()
}
setwd(dir.figures)

#7. Plot module heatmap and eigengene expression####

datME=MEs

# for (module in intModules){ #(see above)
#   which.module=module
#   ME=datME[, paste("ME",which.module, sep="")]
#   par(mfrow=c(2,1), mar=c(0.3, 5.5, 10, 2))
#   #plotDendroAndColors(sampleTree2, traitColors,cex.dendroLabels = 0.5,cex.colorLabels=.5,cex.lab=.7,cex.axis=.7,
#   #                    groupLabels = names(datTraits),saveMar=F)
#   plotMat(t(scale(datExpr[,bwmoduleColors==which.module ][sampleTree2$order,]) ),nrgcols=30,rlabels=F,clabels=row.names(datExpr)[sampleTree2$order],rcols=which.module,ccols=as.character(classcolours[[i]])[sampleTree2$order],main=which.module, cex.main=1)
#   par(mar=c(4, 3.8, 0, 0.7))
#   barplot(ME[sampleTree2$order], col=which.module, main="", cex.main=2,
#           ylab="eigengene expression",xlab="array sample")
#   export.plot(file.prefix=paste('fig',q.i,'.',an.count,'.',i,'.',figure,names(questions)[[q.i]],'.',analysis,names(datalist)[i],module,"dend_order"),export.formats=export.formats.plots,height=height,width=(1 + sqrt(5))/2*height)
#   par(parbackup)
# }
# figure<-figure+1

for (module in intModules){ #(see above)
  which.module=module
```

```
1659   ME=datME[, paste("ME",which.module, sep="")]
1660   par(mfrow=c(2,1), mar=c(0.3, 5.5, 13, 2))
1661   #plotDendroAndColors(sampleTree2, traitColors,cex.dendroLabels = 0.5,cex.
         colorLabels=.5,cex.lab=.7,cex.axis=.7,
1662   #                    groupLabels = names(datTraits),saveMar=F)
1663   plotMat(t(scale(datExpr[,bwmoduleColors==which.module ]) ),nrgcols=30,
         rlabels=F,clabels=row.names(datExpr),rcols=which.module,ccols=as.
         character(classcolours[[i]]),main=which.module, cex.main=1)
1664   par(mar=c(4, 3.8, 0, 0.7))
1665   barplot(ME, col=which.module, main="", cex.main=2,
1666           ylab="eigengene expression",xlab="array sample")
1667   export.plot(file.prefix=paste('fig',q.i,'.',an.count,'.',i,'.',figure,
         names(questions)[[q.i]],'.',analysis,names(datalist)[i],module,"orig_
         order"),export.formats=export.formats.plots,height=height,width=(1 +
         sqrt(5))/2*height)
1668   par(parbackup)
1669 }
1670 figure<-figure+1
1671
1672 par(parbackup)
1673
1674 #boxplots and stats for DE modules
1675 par(parbackup)
1676 for (module in intModules){ #(see above)
1677   which.module=module
1678   ME=datME[, paste("ME",which.module, sep="")]
1679   boxplot(list(ME[tbstat.class[[i]]==levels(tbstat.class[[i]])[[1]]],ME[
         tbstat.class[[i]]==levels(tbstat.class[[i]])[[2]]]),col=c("blue","red"),
         main=module,names=levels(tbstat.class[[i]]))
1680   export.plot(file.prefix=paste('fig',q.i,'.',an.count,'.',i,'.',figure,
         names(questions)[[q.i]],'.',analysis,names(datalist)[i],module,"boxplot"
         ),export.formats=export.formats.plots,height=height,width=(1 + sqrt(5))/
         2*height)
1681 }
1682 figure<-figure+1
1683
1684 #stats for comparisons
1685 statlist.modules<-list()
1686 for (mod in 1:length(intModules)) {
1687   which.module=intModules[[mod]]
1688   ME=datME[, paste("ME",which.module, sep="")]
1689   statlist.modules[[mod]]<-wilcox.test(
1690     ME[tbstat.class[[i]]==levels(tbstat.class[[i]])[[1]]],ME[tbstat.class[[i
       ]]==levels(tbstat.class[[i]])[[2]]]
1691 )$p.value}
1692 tests.modules<-data.frame("names"=intModules,"pval"=as.vector(unlist(
       statlist.modules)),"man_B"=as.vector(unlist(statlist.modules))*length(
       intModules),"bonf"=p.adjust(as.vector(unlist(statlist.modules)),method="
       bonferroni"),"BH"=p.adjust(as.vector(unlist(statlist.modules)),method="
       BH"))
1693 write.csv(tests.modules,file=file.path(dir.results,paste("Module_stats_",q.i
       ,'.',an.count,'.',i,"_",names(datalist)[[i]],names(meth)[[h]],"_",
       colname,"_wilcox.csv",sep="")))
1694
1695 #the end####
1696 #}wgcna for loop
1697
1698 #clean up
1699 rm(TOM)
```

```
collectGarbage()

}

rm(list=as.character(dataset.variables))
collectGarbage()
}

print(paste("End time:",Sys.time()))
```

Listing A.12: SevenQuestions